5-2001

Markov set-chains as models of plant succession

Corey L. Samuels
University of Tennessee, Knoxville

To the Graduate Council:

I am submitting herewith a dissertation written by Corey L. Samuels entitled "Markov set-chains as models of plant succession." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Ecology and Evolutionary Biology.

Mark Kot, Thomas G. Hallam, Major Professor

We have read this dissertation and recommend its acceptance:

Mitch Cruzan, James A. Drake, Robert V. O'Neill, Daniel Simberloff, Jake Weltzin

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

We are submitting herewith a dissertation written by Corey L. Samuels entitled " Markov set-chains as models of plant succession." We have examined the final copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Ecology and Evolutionary Biology.

Mark Kot, Major Professor

Thomas G. Hallam, Department Head

We have read this dissertation
and recommend its acceptance:

Mitch Cruzan

James A. Drake

Robert V. O'Neill

Daniel Simberloff

Jake Weltzin

Accepted for the Council:

Interim Vice Provost and
Dean of the Graduate School

MARKOV SET-CHAINS AS MODELS OF PLANT SUCCESSION

A Dissertation
Presented for the
Doctor of Philosophy Degree
The University of Tennessee, Knoxville

Corey L. Samuels
May, 2001

ACKNOWLEDGEMENTS

First of all, I thank my dissertation advisor, Mark Kot. He has been exceptionally enthusiastic and generous with his time, and I cannot imagine accomplishing this without him. Thank you also to my graduate committee members, Mitch Cruzan, Jim Drake, Bob O'Neill, Dan Simberloff and Jake Weltzin, for their time and input. I am also grateful to Wes Jackson, Jon Piper and David Van Tassel of The Land Institute. Besides contributing grassland data for this project, The Land has been a bountiful source of inspiration, feedback and financial support over the years.

I am overwhelmed to think of the many people who have given me friendship, encouragement, and technical help throughout this process. I am grateful for the daily friendship and collegial interactions of the graduate students and faculty of the University of Tennessee Department of Ecology and Evolutionary Biology. Particularly ever-present friends and colleagues throughout my time at UT were Monica Beals, Matt DeAngelis, Stan Guffey, Michael Kane, Julie Lockwood, Brian Roczen, Chris Skelton, and Carolyn Wells. Many thanks also to Bela Frigyik at UW.

I also thank my friends outside of school, Jennifer Jacoby, Adam Harmon, Dave Kenny, and my brother, Paul Samuels, for patiently listening to daily progress reports, and for doing so much to make my life easier. Finally, I deeply appreciate the support of my phenomenal family; especially my parents, Edward and Stephanie Samuels, who taught me that I can do anything I set my mind to.

This work was supported in part by a Geraldine R. Dodge Foundation Graduate Student Fellowship in Natural Systems Agriculture, and by the National Science Foundation (DMS-9973212).

ABSTRACT

In this dissertation I examine Markov set-chains as a new approach for modeling plant succession. Set-chains are an extension of Markov chains, due to Hartfiel (1991, 1998), that makes it possible to model succession when transition probabilities are uncertain or fluctuating. In Markov set-chains each transition probability is expressed as an interval containing the range of all possible values for that parameter. In turn, a set-chain predicts community composition as a range of possible frequencies for each species. First, I give an introduction to Markov set-chains and methods for iterating and finding their asymptotic behavior. I demonstrate the formulation and computation of a set-chain with an example from a grassland restoration experiment. Next, I use set-chains to investigate the dynamics of experimental grassland plots planted with different species diversities. The set-chain predicts that plots with more planted species will vary less in composition than those with fewer species. I analyze a restricted, two-state set-chain and show that these differences in variability reflect variability thresholds that identify four distinct regions of parameter-space. These regions delineate which transition probability intervals lead to widening, or narrowing, distribution intervals as the system develops. Finally, I use simulations to investigate several questions about how uncertainty propagates from data to parameter estimates and predictions in Markov set-chains. Markov set-chains are an important contribution to our understanding of what controls variability in ecological systems; they may be useful tools for getting more predictable outcomes from ecological restoration and construction.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In ecological theory, models can be used to generate and test ideas about what controls the structure, function and dynamics of communities. Increasingly, models also play a role in applied ecology: practitioners of ecological restoration and management are looking to models as a source for rules or guidelines for ecological management and restoration (Hobbs and Norton 1996, Lockwood 1997, Lockwood and Samuels unpublished manuscript). Indeed, some recent papers have reported on models that address specific technical aspects of restoration (Dobson et al. 1997, Huxel and Hastings 1999). On the applied side, Piper and Pimm (2001, in review) conducted a field experiment to test theoretically derived assembly rules in constructed grassland communities (Piper and Pimm 2001, in review).

It seems clear that there is no single recipe for restoring or constructing a particular community. Findings from models and experiments in community assembly indicate that events early during system assembly can alter the trajectory and outcome of community development. The timing and order of species arrivals may be critical in determining the composition of the resulting community (Robinson & Dickerson 1987, Drake 1991, Drake et al. 1993, Wilbur and Alford 1985). In at least some cases, chance or historical events make it possible for many alternative communities to develop in very similar settings (McCune and Allen 1985, Pickett and Parker 1994, Samuels and Drake,

1997). Additionally, varying the size or composition of the initial species pool can change the trajectory of community development and the composition of the resulting community (Case 1990, Case 1991, Drake 1990, Drake 1991). While it is unlikely that models will ever tell us how to engineer the restoration of a precise, detailed version of the community we want to restore, it may be possible to narrow down the range of possibilities. Markov set-chain models offer a new approach that can help do this.

Markov chains, first introduced into ecology by Horn (1975, 1976) were initially met with great enthusiasm from ecologists. They are simple, depend on transition probabilities, and generate a unique stationary distribution. However, Markov chain models of succession require that transition probabilities be precisely measured and constant for all time. In natural systems, environmental conditions and biotic interactions are dynamic, and transition probabilities are seldom constant. Markov chains were eventually deemed too rigid to be effective predictive models of ecological communities (Usher 1981, Hobbs 1983).

Markov set-chains are an extension of Markov chain models that solve this problem by incorporating uncertainty in their parameters and predictions. Markov set-chains take uncertainty into account, but still deliver quantitative predictions. Set-chains are relatively new to mathematics (Hartfiel 1991, 1998) and until now have not been explored in ecology (Hartfiel 1998). In Markov set-chains, each transition probability parameter is expressed as an interval containing the range of all possible values for that parameter. In turn, a set-chain predicts community composition as a range of possible

frequencies for each species. In this dissertation, I examine and demonstrate the use of Markov set-chains as models of ecological succession.

In each chapter, I draw upon a data set from the grassland restoration experiment at The Land Institute in Salina, Kansas (Piper and Pimm 2001, in review). The experiment compared successional grassland plots that had been planted with different species diversities and were observed over six years. This data set lends itself well to the set-chain approach, and also presents a real-world challenge. The researchers at The Land Institute developed the experiment to help guide the engineering of plant communities for natural systems agriculture. The ultimate goal of their experiment, and what I had in mind for my models, was to learn what species to plant and in what proportions, in order to produce a specific, desired community.

Chapter 2 is an introduction to Markov set-chains as models of succession. I outline and compare three different methods for iterating and finding asymptotic behavior of set-chains. To demonstrate, I formulate and compute distributions from a set-chain from the grassland restoration experiment. Markov set-chains can help us understand what conditions affect predictability and stability of community dynamics; they may be useful tools for managing ecological restoration and construction.

Markov set-chains are new and interesting as models, but what can they tell us about ecology that we did not already know? In Chapter 3, I use a set-chain model to analyze a grassland data set. The model, intriguingly, suggests that more diverse systems will be less variable in composition than less diverse systems. To understand why this might occur, I analyze a restricted, two-state set-chain. I find that there are variability

thresholds that delineate the conditions under which systems become increasingly or decreasingly variable.

The final chapter addresses some of the many questions that arise about implementing Markov set-chains. Set-chains appear quite promising for use in ecology, but this new modeling approach brings with it a whole battery of new questions about how they work and under what conditions they generate useful predictions or are reliable to use. How big can intervals be and still generate meaningful predictions? How much error is associated with estimation in methods for iterating set-chains? I investigate attributes of three sources of error in Markov set-chains, using simulations to consider how each one affects the set-chain predictions.

With the work presented here, I hope to contribute to theoretical and applied ecology. First of all, I contribute to ecological theory by exploring a new modeling approach. Next, my analysis of the model to find variability thresholds contributes to our understanding of the link between diversity and variability. Finally, these variability thresholds make set-chains a useful tool for determining how much diversity is needed to achieve desired outcomes from ecological restoration or construction. This work has opened up a variety of paths for future investigation and application of Markov set-chains.

CHAPTER 2

A NEW APPROACH FOR MODELING SUCCESSION

In this paper I consider a new approach for modeling ecological succession. It is an extension of the Markovian model originally introduced by Horn (1975, 1976). Markov chain models represent succession as a plant-by-plant replacement process. The dynamics depend on the probability of species or functional groups replacing one another in discrete time. A difficulty with applying these models in natural systems is that transition probabilities may be imprecisely measured or fluctuate over time. Markov set-chains solve this problem by defining parameter intervals that contain the range of possible transition probabilities. Somewhat surprisingly, these interval-based models retain many of the properties that make regular Markov chains appealing. Hartfiel (1981, 1987, 1991, 1998) introduced and has developed the mathematical theory of Markov set-chains. He cites few applications, and I am not aware of any examples where this approach has been applied in ecology. Markov set-chain models offer a new way to predict the outcome of succession while allowing for natural fluctuations in community composition and the mechanisms of change.

Markov chain models have had a boom and bust history in ecology. Horn's (1975, 1976) papers led to a flurry of applications. Models of forest succession are most prominent (Binkley 1980, Hubbell and Foster 1987, Stephens and Waggoner 1970, 1980, Waggoner and Stephens 1970), but Markov chain models have also been applied to a

variety of other systems. Among the well-known applications are grasslands (Usher 1981), desert (McAuliffe 1988) and heath (Hobbs 1983, Hobbs and Legg 1983). Markovian models are attractive because they are simple, probabilistic and history-independent. However, by the mid-1980's Markov chains were widely criticized for their inadequacies as models of natural systems (Usher 1981, Hobbs 1983). Markov chains are problematic because transition probabilities must be specified precisely and accurately, and must be stationary in time. It is well known that transition probabilities in natural systems vary from year to year, depending on stochastic environmental factors such as precipitation and disturbance. Further, some margin of error always arises when measuring transition probabilities in the field.

Various approaches have been proposed to solve this problem, but with little concrete success. Inhomogeneous Markov chains use a different matrix of transition probabilities for each time step. These were used by Logofet et al. (1997) to model grassland regeneration. Semi-Markov chains incorporate information about how long an individual must stay in a given state before making a transition. Acevedo (1981) used semi-Markov chains to model tropical forest succession. Hidden Markov chains deal with situations where the process is inhomogeneous, but we do not know which mechanisms are operating when. Each approach has difficulties. Inhomogeneous and hidden Markov chains seldom converge to a stable state. Semi-Markov chain models lead to transition probabilities that are a function of some other factor in the system, taking away the simplicity that makes Markov chains so appealing.

So, why is yet another solution worth exploring? Markov set-chains are fundamentally different from these other approaches in that they accommodate fluctuating or uncertain parameter values. Like regular Markov chains, set-chains converge asymptotically in a manner independent of initial conditions. Additionally, they give an idea of how measurement error propagates over time. Because the solution is expressed as an interval, set-chains also give best- and worst-case scenarios that help us make management decisions. Markov set-chains address a primary criticism of Markov chain models. However they also reinforce the basis for criticism, as uncertainty in parameters can lead to increasing uncertainty of the solutions.

In this paper, I introduce the theory of Markov set-chains. First, I explain the general principles of formulating and iterating set-chains to find the distribution over time and asymptotically, in the limit of large time. There are two different approaches to iterating set-chains. One method is exact but computationally intensive while the other is quick but approximate. Throughout, I illustrate the process and methods with a simple model of grassland succession. The set-chain approach leads to a new perspective on the dynamics and variety of pathways in succession. Finally, I discuss some possible applications of Markov set-chains in ecology. What types of problems and data can Markov set-chains be used for? Some interesting approaches result from focusing on the extremes of what can happen. This initial look at Markov set-chains points out some interesting questions about the relation between the extremes and the most likely.

Markov chains and grasslands

Properties and predictions of regular Markov chains

Markov chains are an attractive way to model succession. They are simple to formulate and compute, and can be iterated to predict how directional community change will proceed. Under most circumstances, Markov chains converge to a unique stationary distribution that is independent of initial conditions. In a Markov chain the community comprises individuals. An individual is denoted here as x. At any time t, each individual is in one of some finite number n of discrete states. The community composition at any time t is given by the vector $\mathbf{x} = [b_1, b_2, \ldots, b_n]$, where b_i is the proportion of individuals in state i. An individual in state i is replaced by one of state j with some characteristic transition probability,

$$a_{ij} = \Pr(x_{t+1} = j \mid x_t = i)\,. \tag{2.1}$$

The transition matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \ddots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \tag{2.2}$$

gives the probability of transitions between all pairs of states. A row represents all possible fates given a particular starting point. The probabilities in each row add up to 1. A column represents all possible ways of arriving at one state.

Numerous sources give in-depth explanations of how regular Markov chains are formulated and iterated. Jeffers (1978), Usher (1992) and Guttorp (1995) all provide nice discussions with emphasis on modeling ecological processes. Here, I only briefly review

the procedures for iterating to find the distribution of the various states over time because it is the basis of the procedure for iterating Markov set-chains.

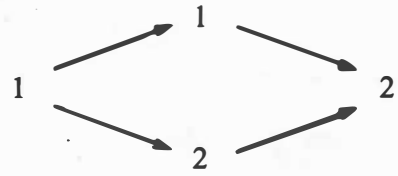
A regular Markov chain can be iterated via matrix multiplication to find the probability of each type of transition occurring in more than one time step. The basic process for iterating is illustrated in Figure 2.1. The transition matrix gives the probability of each replacement occurring in one step. For a system with two states, there are two ways to get from state 1 to state 2 in two steps (Figure 2.1b.) There are four ways to get there in three steps (Figure 2.1c.) The sum of the probabilities of traveling each path gives the two-step (or three-step) value of a_{ij}. Conveniently, the process of figuring out all the different paths and adding them up is equivalent to matrix multiplication. When transition probabilities are constant, the matrix taken to each successive power gives the chance of moving from one state to another in the corresponding number of steps.

Given some initial conditions, two different methods may be used to find the distribution among states at any future time from a regular Markov chain. The first is to take successive powers of the transition matrix **A** as described above and use the relations, $\mathbf{x}_0\mathbf{A} = \mathbf{x}_1$, $\mathbf{x}_0\mathbf{A}^2 = \mathbf{x}_2, \ldots, \mathbf{x}_0\mathbf{A}^n = \mathbf{x}_n$. Alternatively, the solution may be iterated using the distribution at each time, where $\mathbf{x}_0\mathbf{A} = \mathbf{x}_1$, $\mathbf{x}_1\mathbf{A} = \mathbf{x}_2$, $\mathbf{x}_{n-1}\mathbf{A} = \mathbf{x}_n$. The two methods I outline in this paper for iterating and finding asymptotic behavior of Markov set-chains parallel this distinction. The Hi-Lo method is more closely related to the former, whereas the vertex calculation method has its basis in the latter.

(a) One step

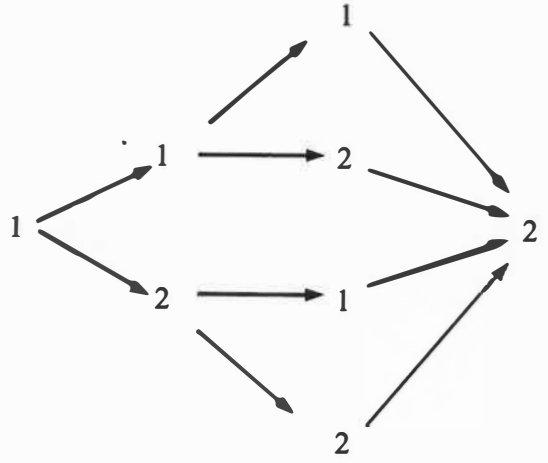
$= p_{12}$

(b) Two steps



$= p_{11}\,p_{12} + p_{12}\,p_{22}$

(c) Three steps



$= p_{11}\,p_{11}\,p_{12} + p_{11}\,p_{12}\,p_{22} + p_{12}\,p_{21}\,p_{12} + p_{12}\,p_{22}\,p_{22}$

Figure 2.1: Calculating Markov chain transitions. p_{ij} is the one-step probability of replacement of an individual in state i by one in state j. As the time interval increases, the number of possible routes between any two states grows.

Grassland restoration as a Markov process

To illustrate this introduction to Markov set-chains, I draw on an example from grassland restoration. Restoration of former agricultural land to prairie is essentially a process of native perennials replacing weedy annuals (Holt et al. 1995). Perennial grasses establish first, followed by herbaceous dicots. Certainly, there is far more to succession than that, but I purposely take this oversimplified view to minimize the matrix size for demonstration. I consider a two-state (annual, perennial) and a three-state (annual, perennial graminoids, other perennials) model at different stages.

The transition matrices are derived from a prairie restoration experiment conducted at The Land Institute (TLI) in central Kansas. Native perennials were seeded on former agricultural land in four replicate plots. Species composition was recorded on all plots annually in July 1994-1999. Piper and Pimm (2001, in review) give a full account of the experiment. From the five years of observation it is clear that a process of directional replacement is occurring. In established prairie, annuals represent less than 1 percent of cover (Piper, 1995). In the plots at TLI, initially up to 70 percent of plant cover was annual. By 1999 all plots showed an increase in perennials to 85-95 percent of cover (Table 2.1).

As a general trend, this successional process is not particularly remarkable or unexpected. However, variation in the dynamics from plot to plot may generate fluctuations in the composition of the system. Figure 2.2 shows the Markov transition matrices for each plot, along with the predicted stable distribution calculated from each one. A unique community structure is predicted for each plot. The variation is not huge,

Table 2.1: Proportion of biomass represented by each of (a) two functional groups and (b) three functional groups in four replicate plots during years 1994-1999. Group codes are a = annuals, p = perennials, g = perennial graminoids, o = other perennials.

Year	Plot 1		Plot 2		Plot 3		Plot 4	
(a)								
	a	p	a	p	a	p	a	p
1994	0.692	0.308	0.682	0.318	0.567	0.433	0.522	0.478
1995	0.315	0.685	0.461	0.539	0.327	0.673	0.358	0.642
1996	0.226	0.774	0.309	0.691	0.263	0.737	0.119	0.881
1997	0.075	0.925	0.037	0.963	0.063	0.937	0.106	0.894
1998	0.020	0.980	0.013	0.987	0.029	0.971	0.062	0.938
1999	0.070	0.930	0.033	0.967	0.127	0.873	0.134	0.866
(b)								
	a	g	o	a	g	o	a	g	o	a	g	o
1994	0.692	0.082	0.226	0.682	0.042	0.276	0.567	0.129	0.304	0.522	0.314	0.164
1995	0.315	0.194	0.492	0.461	0.072	0.467	0.327	0.207	0.466	0.358	0.149	0.493
1996	0.226	0.318	0.456	0.309	0.213	0.478	0.263	0.327	0.411	0.119	0.233	0.648
1997	0.075	0.453	0.472	0.037	0.193	0.77	0.063	0.414	0.523	0.106	0.281	0.613
1998	0.020	0.244	0.735	0.013	0.150	0.837	0.029	0.299	0.672	0.062	0.148	0.790
1999	0.070	0.358	0.572	0.033	0.231	0.736	0.127	0.284	0.589	0.134	0.163	0.703

(a) 2x2 Matrices for plots 1-4.

$$\mathbf{A}_3 = \begin{array}{c} \begin{matrix} a & p \end{matrix} \\ \begin{bmatrix} .442 & .558 \\ .033 & .967 \end{bmatrix} \end{array}, \quad \mathbf{A}_2 = \begin{array}{c} \begin{matrix} a & p \end{matrix} \\ \begin{bmatrix} .567 & .433 \\ .009 & .991 \end{bmatrix} \end{array},$$

$$\mathbf{A}_3 = \begin{bmatrix} .482 & .518 \\ .057 & .943 \end{bmatrix}, \quad \mathbf{A}_4 = \begin{bmatrix} .479 & .521 \\ .059 & .941 \end{bmatrix}.$$

Predicted stable distribution

$\mathbf{s}_1 = (.056, .944)$, $\mathbf{s}_2 = (.020, .980)$,
$\mathbf{s}_3 = (.099, .901)$, $\mathbf{s}_4 = (.102, .898)$.

(b) 3x3 Matrices for plots 1-4.

$$\mathbf{A}_1 = \begin{array}{c} \begin{matrix} a & g & o \end{matrix} \\ \begin{bmatrix} .436 & .147 & .416 \\ 0 & .132 & .868 \\ .055 & .505 & .440 \end{bmatrix} \end{array}, \quad \mathbf{A}_2 = \begin{array}{c} \begin{matrix} a & g & o \end{matrix} \\ \begin{bmatrix} .565 & .080 & .355 \\ 0 & 0 & 1 \\ .012 & .261 & .727 \end{bmatrix} \end{array},$$

$$\mathbf{A}_3 = \begin{bmatrix} .467 & .233 & .301 \\ 0 & .443 & .557 \\ .099 & .263 & .638 \end{bmatrix}, \quad \mathbf{A}_4 = \begin{bmatrix} .468 & .234 & .298 \\ .081 & .039 & .879 \\ .055 & .241 & .704 \end{bmatrix}.$$

Predicted stable distribution

$\mathbf{s}_1 = (.057, .353, .590)$, $\mathbf{s}_2 = (.021, .204, .775)$,
$\mathbf{s}_3 = (.107, .317, .576)$, $\mathbf{s}_4 = (.102, .200, .698)$.

Figure 2.2: Transition matrices and predicted stable distribution for four grassland restoration plots for (a) two functional groups and (b) three functional groups. Functional groups (states) are a = annuals, p = perennials, g = perennial graminoids, and o = other perennials. There is considerable variation among Markov chains and predictions for the four replicates.

but it might reflect the difference between achieving a native-like state and never quite getting there. In Piper's (1995) study of prairie composition at The Land Institute, 95 percent perennial cover is not far from usual, but 85 percent never happens. How important is the variation in transition probabilities from year to year or plot to plot? Markov set-chains can be used to study the breadth of possibilities when transition probabilities vary.

Setting up the Markov set-chain.

In a Markov set-chain, each transition probability is replaced by an interval defined by the minimum and maximum possible probability of replacement. For instance, in the grasslands at TLI in any given plot and year, anywhere from 40 to 60 percent of the annual cover is replaced by perennials. The transition probability is the interval (.4, .6). Figure 2.3 illustrates graphically the difference between the stationary transition probability points and Markov set-chain intervals. Most notably, instead of a single point, the interval defines the range of combinations of probabilities of arriving at each state. The set of probabilities (a row of the matrix) must each be points on their respective intervals and together make a row totaling 1. The transition probability intervals are expressed in two separate matrices,

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix} \text{ and } \mathbf{Q} = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1n} \\ q_{21} & q_{22} & \cdots & q_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ q_{n1} & q_{n2} & \cdots & q_{nn} \end{bmatrix} \tag{2.3}$$

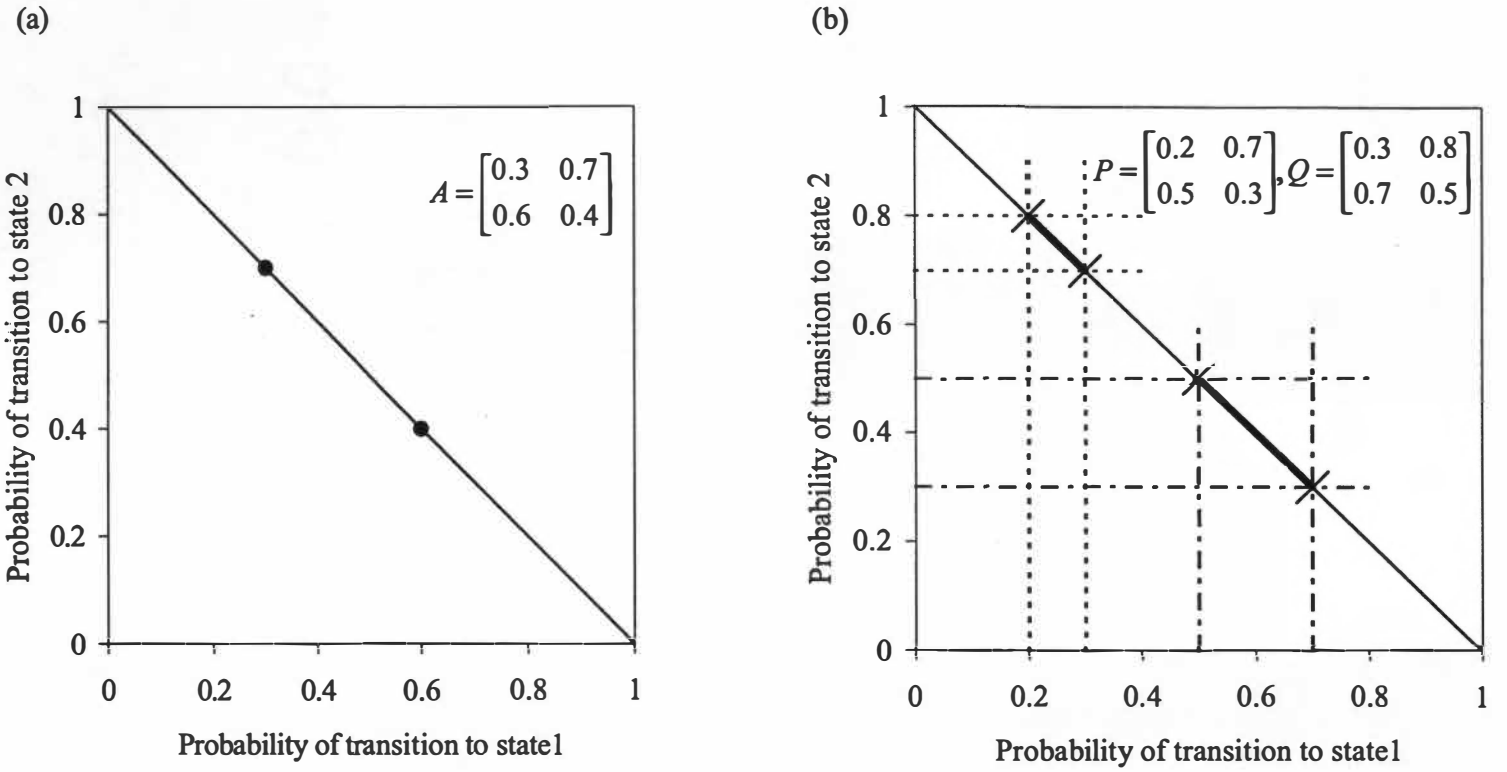


Figure 2.3: Graphical comparison of (a) transition probability points in a regular Markov chain and (b) transition probability intervals in a Markov set-chain. P is the matrix of all the lowest possible transition probabilities and Q contains the highest. Because this is a two-state system, all probabilities lie on the line where the sum of transitions to the two states is 1.

where p_{ij} is the smallest chance and q_{ij} is the largest chance of making the transition from state i to state j in one step. Note that the rows of the **P** and **Q** matrix do not sum to 1. In any given year any value within the interval, including the minimum or maximum, could occur. However, the combinations of parameters that can co-occur are limited. A row of the *actual* matrix **A** of transition probabilities for any given year represents, for an individual in one state, the chance of replacement by individuals of every other state in that particular year. All parameters in the row are drawn from within their respective intervals and the elements of the row do add up to 1. This limits the combinations of parameter values in that, for example, every parameter cannot achieve its maximum value in the same year.

The **P** and **Q** matrices of the grassland are, for the 2-state system,

$$\mathbf{P} = \begin{bmatrix} 0.442 & 0.433 \\ 0.009 & 0.941 \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} 0.567 & 0.558 \\ 0.059 & 0.991 \end{bmatrix} \tag{2.4}$$

and for the 3-state system,

$$\mathbf{P} = \begin{bmatrix} .436 & .080 & .298 \\ 0 & 0 & .557 \\ .012 & .242 & .440 \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} .565 & .234 & .416 \\ .081 & .443 & 1 \\ .099 & .505 & .727 \end{bmatrix}. \tag{2.5}$$

Making sure interval boundaries are tight

Each pair of states i and j has an interval $[p_{ij}, q_{ij}]$ of possible transition probabilities. The minimum and maximum of each interval must be tested, and adjusted if necessary, to ensure that it is *tight*. A boundary is tight if it combines exactly (tightly) with other parameters from row i of **P** and **Q** to make a row that sums to 1. For example,

Figure 2.4a represents one row an imaginary two state system where the transition probability intervals are defined by **p** = (0.2, 0.4) and **q** = (0.6, 0.7). Specified individually, each probability interval is perfectly reasonable. An individual arrives at state 1 between 20 and 60 percent of the time. It arrives in state 2 from 40 to 70 percent of the time. Each element of **p** and **q** must be checked individually.

I begin with p_{11}. Imagine a year where p_{11} happens to take on its smallest possible value, 0.2. What happens to the rest of the individuals? Since there are only two states, if 0.2 remain in state 1, the other 0.8 must then be replaced by state 2. However, looking to the maximum specified for q_{12}, I see that the transitions to state 2 cannot exceed 0.7. So the interval must be tightened by adjusting $p_{11} = 0.3$. There is no need to adjust q_{12}. Imagine a year where 0.7 make the transition to state 2. This requires that 0.3 remain in state 1. Since 0.2 is the minimum, and 0.3 is within the specified interval, (0.2, 0.6), $q_{11} = 0.7$ need not be adjusted. More formally, the new tightened minimum value a_{ij} for each parameter must be

$$\min a_{ij}$$

$$\text{such that } \sum_{k \neq j} a_{ik} + a_{ij} = 1 \tag{2.6}$$

$$\text{for some } a_{ik}, a_{ik} \in [p_{ik}, q_{ik}].$$

A similar definition holds for max a_{ij}. All parameters should first be tested for tightness using the originally specified interval boundaries, and then necessary adjustments made all at once afterwards.

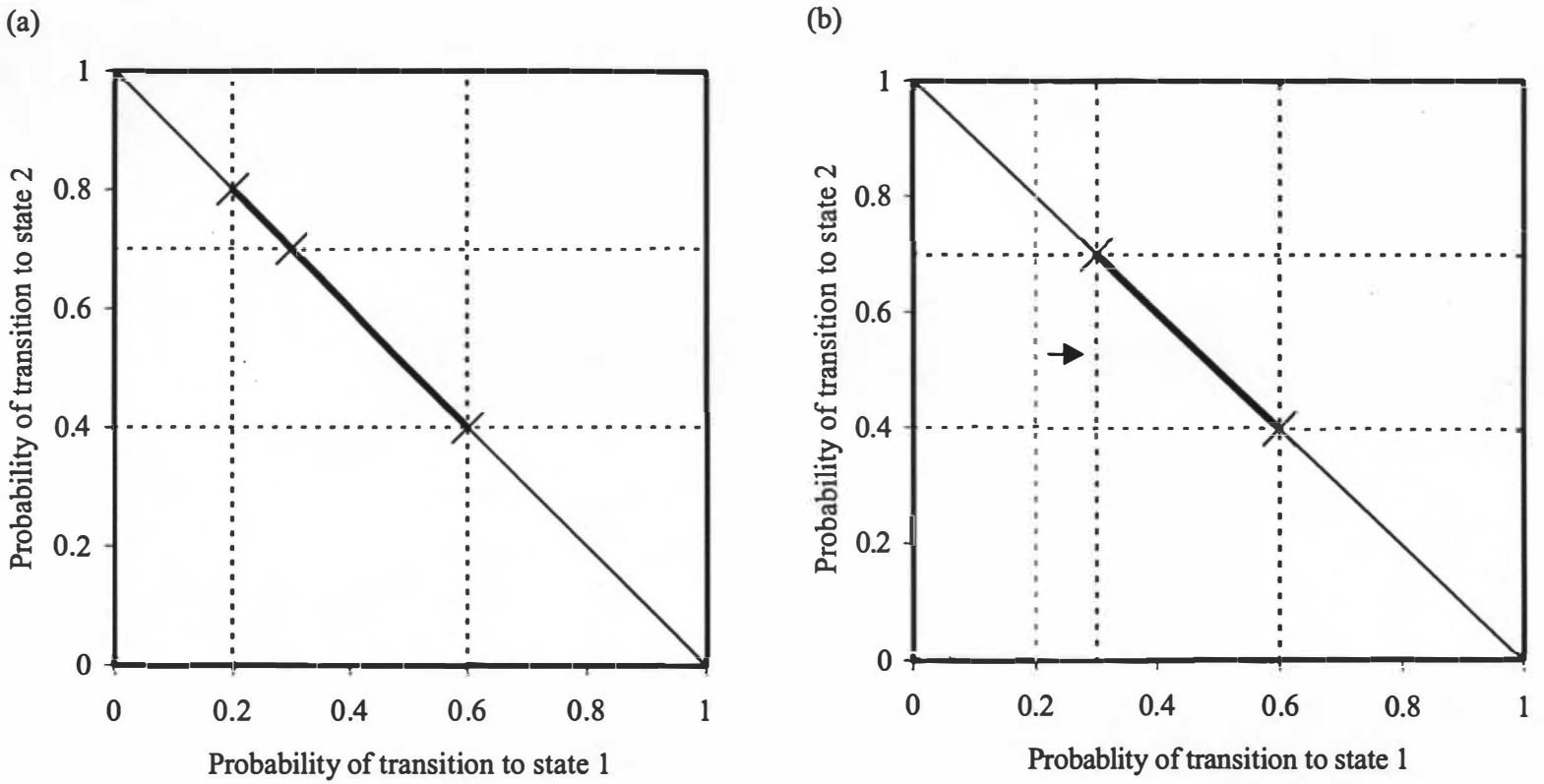


Figure 2.4: Tightening the interval for a row of a two-state matrix. Transition probability intervals are defined by $p = (0.2, 0.4)$ and $q = (0.6, 0.7)$. (a) shows the original 'loose' interval. (b) shows the tightened interval.

Often, tightness is not a major concern. When data are derived directly from a set of observations, where the sum of all observations is 1, intervals are automatically tight. It is more likely that adjustments to intervals to guarantee tightness will be required when data are estimated or intervals come from error estimates. This may occur in a case where we know the average transition probabilities, but want to allow for 5% error in measurement or fluctuation from year to year. Table 2.2 shows the tests for tightness done on the 2x2 and 3x3 grassland examples.

Iterating and solving for boundaries on behavior over time

The tight interval gives the range of possibility for each type of replacement occurring during one step in time. Now, we want to use these one-step transition probability intervals to calculate the behavior and composition of the system at successive times. In this section I explain three different ways of using the set-chain to predict community dynamics: calculating vertices, the Hi-Lo method (Hartfiel 1991, 1998) and Monte Carlo simulation. The first two are explained mathematically in Hartfiel (1987, 1991, 1998). There are tradeoffs among the three as to computational effort, accuracy of predictions, and asymptotic behavior (Table 2.3.) I give a brief explanation of how each method works, and then demonstrate using the grassland example.

Table 2.2: Testing intervals for tightness in Markov set-chain model of grassland restoration for (a) 2x2 system and (b) 3x3 system.

Parameter					Test	Tight?
(a)						
p_{11}	0.4422 +	0.5578		= 1	≥1?	yes
p_{12}	0.4330 +	0.5670		= 1	≥1?	yes
p_{21}	0.0086 +	0.9914		= 1	≥1?	yes
p_{22}	0.9408 +	0.0592		= 1	≥1?	yes
q_{11}	0.5670 +	0.4330		= 1	≤1?	yes
q_{12}	0.5578 +	0.4422		= 1	≤1?	yes
q_{21}	0.0592 +	0.9408		= 1	≤1?	yes
q_{22}	0.9914 +	0.0086		= 1	≤1?	yes
(b)						
p_{11}	0.4363 +	0.2342 +	0.4163	= 1.0868	≥1?	yes
p_{12}	0.0800 +	0.5648 +	0.4163	= 1.0611	≥1?	yes
p_{13}	0.2976 +	0.5648 +	0.2342	= 1.0966	≥1?	yes
p_{21}	0 +	0.4432 +	1	= 1.4432	≥1?	yes
p_{22}	0 +	0.0813 +	1	= 1.0813	≥1?	yes
p_{23}	0.5568 +	0.0813 +	0.4432	= 1.0813	≥1?	yes
p_{31}	0.0124 +	0.5048 +	0.7270	= 1.2442	≥1?	yes
p_{32}	0.2415 +	0.0987 +	0.7270	= 1.0672	≥1?	yes
p_{33}	0.4398 +	0.0987 +	0.5048	= 1.0433	≥1?	yes
q_{11}	0.5648 +	0.0800 +	0.2976	= 0.9424	≤1?	yes
q_{12}	0.2342 +	0.4363 +	0.2976	= 0.9681	≤1?	yes
q_{13}	0.4163 +	0.4363 +	0.0800	= 0.9326	≤1?	yes
q_{21}	0.0813 +	0 +	0.5568	= 0.6381	≤1?	yes
q_{22}	0.4432 +	0 +	0.5568	= 1	≤1?	yes
q_{23}	1 +	0 +	0	= 1	≤1?	yes
q_{31}	0.0987 +	0.2415 +	0.4398	= 0.7800	≤1?	yes
q_{32}	0.5048 +	0.0124 +	0.4398	= 0.9570	≤1?	yes
q_{33}	0.7270 +	0.0124 +	0.2415	= 0.9809	≤1?	yes

Table 2.3: Tradeoffs among three methods for solving Markov set-chains.

	Computational effort	Bounds	Typical asymptotic behavior
Vertex calculation	HIGH! Number of vertices to be evaluated increases very quickly. Intractable for large systems or long times.	EXACT.	CONVERGENT. However, it is difficult to obtain solutions for a long enough time to see it.
Hi-Lo algorithm	LOW.	APPROXIMATE.	CONVERGENT.
Monte Carlo simulation	VERY LOW. Solution is by matrix multiplication.	INFERRED. More simulation runs lead to increasing accuracy.	NOT CONVERGENT. Individual runs never converge. However, boundaries on asymptotic behavior may be inferred from many runs.

Iterating by computing the vertices

The set of all matrices **A** with elements on the intervals (p_{ij}, q_{ij}) and rows that sum to 1 may be expressed as a matrix interval **[P, Q]**. Each row of **[P, Q]** may be visualized geometrically. Fig. 3b illustrates the geometric interval for the 2x2 case where

$$\mathbf{P} = \begin{bmatrix} 0.2 & 0.7 \\ 0.5 & 0.3 \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} 0.3 & 0.8 \\ 0.7 & 0.5 \end{bmatrix}. \tag{2.7}$$

Each dark segment in the figure represents one row. The *vertices* for each row are the endpoints of the segment. These endpoints represent the extremes of all possible combinations of transition probability values (i.e., where at least one parameter is either maximized or minimized.) The vertices are (0.2, 0.8) and (0.3, 0.7) for the first row and (0.5, 0.5), and (0.7, 0.3) for the second. The vertices of the *matrix* comprise all possible combinations of vertices of the first row with vertices of the second. Here,

$$\mathbf{E}_1 = \begin{bmatrix} 0.2 & 0.8 \\ 0.5 & 0.5 \end{bmatrix}, \mathbf{E}_2 = \begin{bmatrix} 0.2 & 0.8 \\ 0.7 & 0.3 \end{bmatrix}, \mathbf{E}_3 = \begin{bmatrix} 0.3 & 0.7 \\ 0.5 & 0.5 \end{bmatrix}, \mathbf{E}_4 = \begin{bmatrix} 0.3 & 0.7 \\ 0.7 & 0.3 \end{bmatrix} \tag{2.8}$$

are the four matrix vertices.

The distribution of individuals among states at time t may also be expressed as an interval [r, s], this time in distribution probability space (Hartfiel 1998). The vertices of [r, s] may be used to calculate the vertices of the distribution interval at t+1. If **r** = [0.1, 0.6], and **s** = [0.4, 0.9] then the vertices are $v_1^{(0)}$ = (0.1, 0.9), $v_2^{(0)}$ = (0.4, 0.6). The product of each matrix vertex and each distribution space vertex gives a point on the distribution interval at the next time step. Each of these points is a potential vertex of the next-step distribution.

The potential vertices are $\mathbf{v}_1^{(0)}\mathbf{E}_1$ = (0.47, 0.53), $\mathbf{v}_2^{(0)}\mathbf{E}_1$ = (0.38, 0.62), $\mathbf{v}_1^{(0)}\mathbf{E}_2$ = (0.65, 0.35), $\mathbf{v}_2^{(0)}\mathbf{E}_2$ = (0.5, 0.5), $\mathbf{v}_1^{(0)}\mathbf{E}_3$ = (0.48, 0.52), $\mathbf{v}_2^{(0)}\mathbf{E}_3$ = (0.42, 0.58), $\mathbf{v}_1^{(0)}\mathbf{E}_4$ = (0.66, 0.34), $\mathbf{v}_2^{(0)}\mathbf{E}_4$ = (0.54, 0.46). Considering these eight points, all lie on the same line, so the interval containing all possible distributions of individuals among states at time t = 1 can be defined by the two endpoints, (0.38, 0.62) and (0.66, 0.34). Next, taking the vertices at $t+1$ as **r** and **s** the process may be repeated to find the distribution at $t+2$ and so on.

For higher dimensions, the distribution space is geometrically more complex. The set of all possible next-step distributions is a volume in n-dimensional distribution space. This volume is the convex hull of all the potential vertices. The matrix interval **[P, Q]** of an n-state system may have up to $(2^{n-1})^n$ vertices. This means even in a relatively simple 3x3 system, **[P, Q]** can have up to 64 vertices. If the initial conditions set has three vertices, calculating the distribution for just the first time step initially generates 192 vertices. A worst-case scenario would have us evaluating nearly four million vertices to find the distribution space after only ten iterations. Thankfully, the convex hull of the potential vertices can often be described by a smaller set of true vertices. For example, if the convex hull is a cube, any potential vertex located inside the cube is not needed.

A variety of algorithms are available for finding the convex hull from a list of points (O'Rourke, 1998). Still, even reducing the list of vertices this way, for a system with more than three states vertex calculation quickly becomes intractable. When it is feasible, however, the advantage of calculating vertices is that the result is a description

of the boundaries on the distribution of states at any time and eventually, asymptotically. In summary, the procedure for calculating the vertices is:

In summary, the procedure for calculating the vertices is:

- Find the vertices of the transition matrix interval.
- Find the vertices of the distribution interval.
- Calculate the products of all transition matrix vertices and distribution interval vertices to obtain potential next-step distribution vertices.
- Find the true vertices of the convex hull of all potential vertices to obtain a minimized list of vertices for the next time step.

Finding the vertices in the grasslands example

In the 2x2 prairie system the interval **[P,Q]** is defined as

$$\mathbf{P} = \begin{bmatrix} 0.442 & 0.433 \\ 0.009 & 0.941 \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} 0.567 & 0.558 \\ 0.059 & 0.991 \end{bmatrix} \tag{2.9}$$

We have already checked and found that the intervals are tight. The vertices of [P,Q] are

$$\mathbf{E}_1 = \begin{bmatrix} 0.442 & 0.558 \\ 0.009 & 0.991 \end{bmatrix}, \mathbf{E}_2 = \begin{bmatrix} 0.442 & 0.558 \\ 0.059 & 0.941 \end{bmatrix}$$
$$\mathbf{E}_3 = \begin{bmatrix} 0.567 & 0.433 \\ 0.009 & 0.991 \end{bmatrix}, \mathbf{E}_4 = \begin{bmatrix} 0.567 & 0.433 \\ 0.059 & 0.941 \end{bmatrix}. \tag{2.10}$$

The initial conditions range is

$$\mathbf{r} = (0.522, 0.308), \mathbf{s} = (0.692, 0.478). \tag{2.11}$$

The vertices of the interval [**r**, **s**] are $\mathbf{v}_1^{(0)} = (0.522, 0.478)$ and $\mathbf{v}_2^{(0)} = (0.692, 0.380)$.

So, to find the vertices of the distribution space at time $t = 1$, we calculate,

$$\begin{aligned}
&\mathbf{v}_1^{(0)}\mathbf{E}_1 = (0.2350, 0.7650),\ \mathbf{v}_2^{(0)}\mathbf{E}_1 = (0.3086, 0.6914),\\
&\mathbf{v}_1^{(0)}\mathbf{E}_2 = (0.2589, 0.7411),\ \mathbf{v}_2^{(0)}\mathbf{E}_2 = (0.3240, 0.6760),\\
&\mathbf{v}_1^{(0)}\mathbf{E}_3 = (0.3003, 0.6997),\ \mathbf{v}_2^{(0)}\mathbf{E}_3 = (0.3951, 0.6049),\\
&\mathbf{v}_1^{(0)}\mathbf{E}_4 = (0.3242, 0.6758),\ \mathbf{v}_2^{(0)}\mathbf{E}_4 = (0.4105, 0.5895).
\end{aligned} \tag{2.12}$$

There are eight vertices, but on examination, the minimum and maximum vertices are $\mathbf{v}_1^{(1)} = (0.2350, 0.7650)$ and $\mathbf{v}_2^{(1)} = (0.4105, 0.5895)$. So, the distribution interval at $t = 1$ is $\mathbf{r} = (0.2350, 0.5895)$, $\mathbf{s} = (0.4105, 0.7650)$. This means that after one year, we have gone from 50-69% annuals to 24-41% annuals. The distribution interval at $t = 2$ can be found by repeating the process, now taking the vertices of the $t = 1$ distribution in place of those for $t = 0$. The vertices of the distribution space iterated over time are shown in Figure 2.5.

I have taken a 2x2 example to demonstrate the method for finding the vertices because it is simple. The interval of a 2x2 system never has more than two vertices because the solution is always a line segment. Any potential vertices other than the minimum and maximum points are be located on the line between the vertices and do not add any information about the solution space. However, this two-vertex example hides the primary difficulty with the vertex calculation method, the rapidly proliferating vertices. Table 2.4 is the list of vertices of the 3x3 system at points during the first five time steps. Even with a simple 3-state system the number of vertices to consider grows with daunting speed. By only the fifth time step the solution space has fourteen vertices.

(a)

Year	Proportion			
	Annuals	Perennials	Annuals	Perennials
1	0.2349	0.5894	0.4106	0.7651
2	0.1105	0.7323	0.2677	0.8895
3	0.0565	0.8049	0.1951	0.9435
4	0.0331	0.8417	0.1583	0.9669
5	0.0230	0.8604	0.1396	0.9770
∞	0.0152	0.8797	0.1203	0.9848

(b)

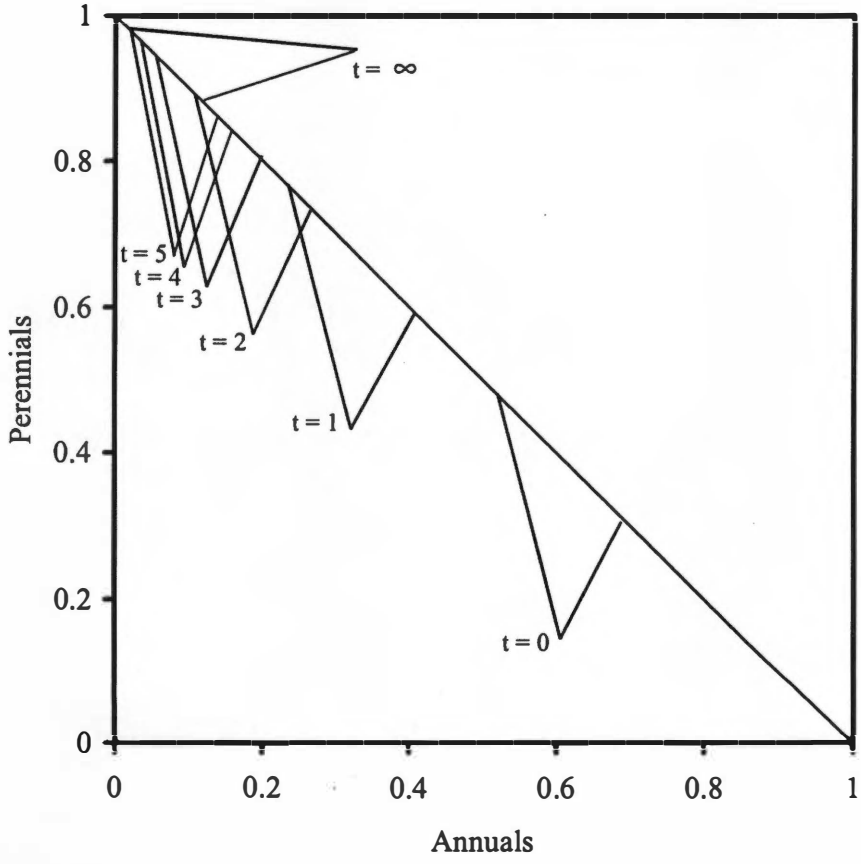


Figure 2.5: Frequency of annual and perennial plant cover found by computing vertices. (a.) Listing of vertices for times 0 - 5 and ∞ . (b.) Geometric representation of relative frequency of annual and perennial plant cover. The interval boundaries change at each time step, approaching their respective asymptotic values. Interval size, the distance between any minimum and maximum boundary, decreases over time.

Table 2.4: List of vertices of the community distribution state-space. By the fifth time step, it takes 14 vertices to define the solution space. Columns and rows represent a=annuals, g=perennial graminoids, o=other perennials.

		a	g	o
t=0	1	1	0	0
	2	0	1	0
	3	0	0	1
t=1	1	0.2298	0.1197	0.6505
	2	0.2298	0.3442	0.4260
	3	0.2315	0.3528	0.4157
	4	0.2612	0.3528	0.3859
	5	0.2650	0.0845	0.6505
	6	0.2791	0.0814	0.6395
	7	0.3365	0.0814	0.5821
	8	0.3387	0.3150	0.3463
	9	0.4205	0.2332	0.3463
	10	0.4205	0.1196	0.4599
t=3	1	0.0519	0.1388	0.8093
	2	0.0519	0.4503	0.4978
	3	0.0529	0.1349	0.8122
	4	0.0553	0.4673	0.4774
	5	0.0689	0.1189	0.8122
	6	0.0876	0.4673	0.4451
	7	0.0910	0.4673	0.4417
	8	0.1244	0.1129	0.7627
	9	0.1682	0.4167	0.4151
	10	0.1957	0.1129	0.6914
	11	0.2326	0.3523	0.4151
	12	0.2326	0.1593	0.6082
t=5	1	0.0189	0.1367	0.8445
	2	0.0189	0.4685	0.5127
	3	0.0189	0.1392	0.8420
	4	0.0203	0.1312	0.8485
	5	0.0229	0.4885	0.4886
	6	0.0317	0.1198	0.8485
	7	0.0597	0.4885	0.4518
	8	0.0789	0.1131	0.8081
	9	0.0868	0.1131	0.8002
	10	0.1192	0.4518	0.4289
	11	0.1515	0.1131	0.7354
	12	0.1759	0.3952	0.4289
	13	0.1911	0.3789	0.4301
	14	0.1911	0.1685	0.6404

Iterating by the Hi-Lo method

The Hi-Lo method offers a way of estimating the boundaries of the distribution space without sifting through hundreds of vertices. It is an algorithm that finds the smallest box containing all the vertices. The solution space, then, is defined by only $2n$ points, regardless of how many times the system is iterated. Both computationally and geometrically, the Hi-Lo method is far simpler than calculating the vertices. The cost of this simplicity is that some degree of estimation is involved, so that bounds are not as accurate. Figure 2.6 shows a comparison of the probability distribution space description obtained from these two methods for the grassland set-chain.

The Hi-Lo method is a modified matrix multiplication approach. Recall that a regular Markov chain is iterated by matrix multiplication. It is not directly possible to do multiplication on an interval matrix, since each parameter can take on any value within its range. Hartfiel's algorithm is a method for constructing rows and columns to multiply together to obtain the minimum and maximum value for each element of the next-step matrix.

The initial **P** and **Q** matrices comprise the minimum and maximum probability of each type of transition occurring in one time step. The aim of the Hi-Lo method is to find the minimum and maximum chance the transition will occur in two (or more) steps. Given a pair of matrices, **P** (low) and **Q** (high), each element of the next-step matrix is computed individually. As in matrix multiplication, each element is found by multiplying a row (the chance of leaving a single state and arriving at each respective other) by a column (the chance of leaving any state and arriving at a single destination.)

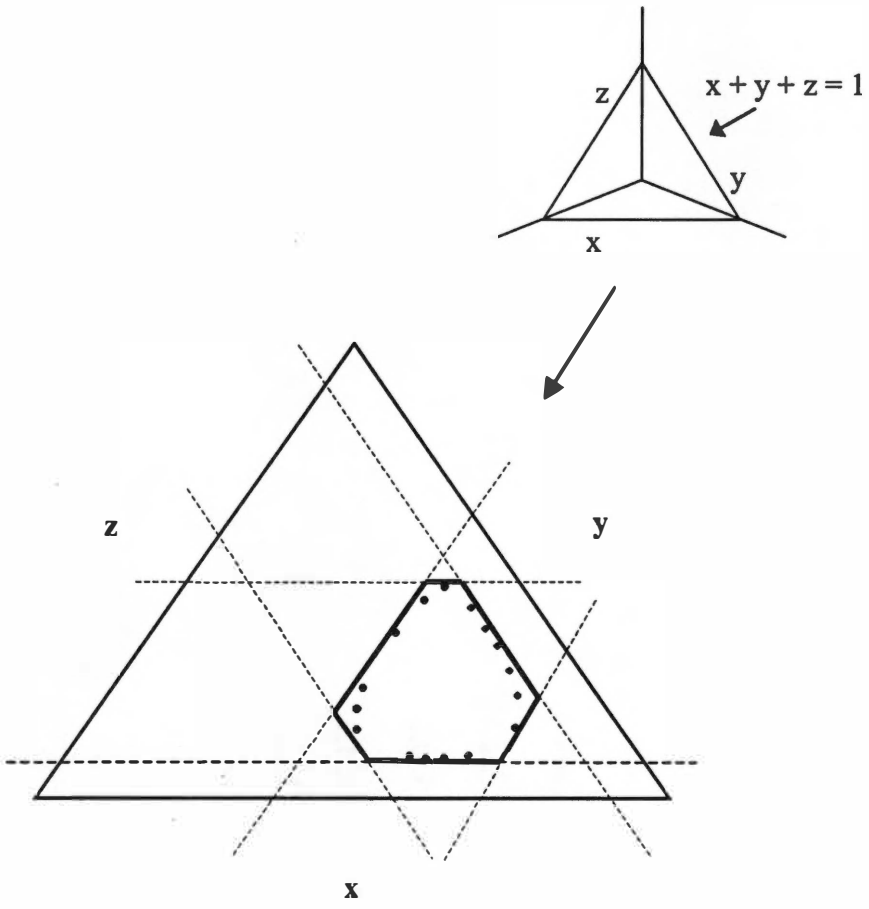


Figure 2.6: Comparison between Hi-Lo approximation and calculated vertices for the interval containing the state space at some time t for a Markov set-chain. Bold lines enclose the interval estimated by the Hi-Lo method. Calculated vertices are shown as points. The Hi-Lo method provides an estimate of the convex hull of the vertices.

The algorithm to find the minimum (or maximum) possible product of column and row is proceeds as follows:

1. To minimize the two-step p_{ij},
   a. Take column j directly from the **P** (low) matrix. Each element of the column is the lowest possible probability of arriving at state j starting from state i.
   b. Construct a row vector made up of values from rows i of **P** and **Q** such that the product of the constructed row and column is minimized. We need to maximize the probability of arriving at states where the transition to j is least likely:
      - Rank the elements of column j of **P** in order from lowest to highest. These rankings dictate the order to assign elements in the constructed row vector.
      - Begin by taking row i of **P** (all minimum transition probabilities) as the constructed row.
      - Set the lowest-ranked element of the constructed row to the value from the corresponding position in row i of **Q** (the largest possible entry).
      - Compute the row sum of the constructed row. If the total is less than 1, set the next-lowest ranked row element to the value from its corresponding position in row i of Q.
      - Repeat the previous step, checking the sum and drawing values until the sum of the row exceeds 1.
      - Finally, take the *last* **Q** value that was added (the one that caused the row sum to exceed 1) and replace it with a value such the entire row now sums to 1.
   c. Multiply the row and the column to find the two-step p_{ij}.

2. To maximize the two-step q_{ij},

   a. Take column j directly from the **Q** (high) matrix. Each element of the column is the highest possible probability of arriving at state j starting from state i.

   b. Construct a row vector made up of values from rows i of **P** and **Q** such that the product of the row and column is maximized. We need to maximize the probability of arriving at states where the transition to j is most likely:

      - Rank the elements of column j of **P** in order from lowest to highest. These rankings dictate the order for assigning elements in the constructed row vector.
      - Begin by taking row i of **Q** (all maximum transition probabilities) as the constructed row.
      - Set the lowest-ranked element of the constructed row to the value from the corresponding position in row i of **P** (the smallest possible entry).
      - Compute the row sum of the constructed row. If the total is greater than 1, set the next-lowest ranked row element to the value from its corresponding position in row i of P.
      - Repeat the previous step, checking the sum and drawing values until the sum of the row is less than 1.
      - Finally, take the *last* **P** value that was added (the one that caused the row sum to exceed1) and replace it with a value such that the entire row now sums to 1.

   c. Multiply the row and the column to find the two-step q_{ij}.

Using the Hi-Lo method to iterate the grassland set-chain

Since the Hi-Lo method is simpler than finding the vertices, we can use the three-state system to demonstrate. Here, the rows and columns represent annuals, perennial grasses and other perennials. The minimum and maximum matrices are

$$\mathbf{P} = \begin{bmatrix} .436 & .080 & .298 \\ 0 & 0 & .557 \\ .012 & .242 & .440 \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} .565 & .234 & .416 \\ .081 & .443 & 1 \\ .099 & .505 & .727 \end{bmatrix}. \quad (2.13)$$

The initial conditions interval is

$$\mathbf{r} = [.522 \quad .042 \quad .164], \mathbf{s} = [.692 \quad .314 \quad .304] \quad (2.14)$$

We have already checked for tightness (Table 2.2) and no adjustments were required.

Let's say we want to find $p_{11}^{(2)}$, the minimum two-step probability of starting in state 1 and ending up in state 1. We need to multiply a row by a column. The row gives the chance of leaving state 1 and arriving at each possible state in one step. The column gives the second step chance of leaving from any of those states and returning to state 1. Since the aim is to minimize, we take the minimum column from **P**, $(0.436, 0, 0.012)^T$. where T denotes the transpose. Choosing the row is trickier. We now must construct a row vector that minimizes the product of row and column. It is not possible to simply multiply by the row from **P** because the row (0.436, 0.080, 0.298) adds up to less than 1.

Ranking the column elements in order from highest to lowest gives $(3, 1, 2)^T$. We replace the row element corresponding to the lowest ranking, element 2, with its maximum value from the **Q** matrix. We now have (0.436, 0.234, 0.298), which has a row sum of 0.968. Since the sum is still less than one, we now replace the next-ranked row elements to get (0.436, 0.234, 0.416). Now the row sum is 1.086. Since the row sum is

now greater than one, we replace the last **Q** added with 0.330 to make the row (0.436, 0.234, 0.330) sum to 1. To find $p_{11}^{(2)}$ we multiply

$$(.436, .234, .330)\begin{pmatrix} .436 \\ 0 \\ .012 \end{pmatrix} = .194. \tag{2.15}$$

In a similar vein, now let's find a maximum, $q_{23}^{(2)}$. The column that gives the maximum chance of arriving at state 3 from all the others is the column from Q, $(0.416, 1, 0.727)^{T}$. Ranking the elements from lowest to highest gives $(1, 3, 2)^{T}$. To construct a row, begin with the row of **Q** that gives the maximum chance of going from state 2 to all others, (0.081, 0.443, 1). Clearly, the sum is more than 1. The smallest element in the column is in position 1, so we replace that one with its value from **P** to get (0, 0.443, 1), still totaling more than 1. Now, the next smallest element is in position 3, so we replace that element to get (0, 0.443, 0.557), which sums to exactly one. No further adjustment is necessary and

$$q_{23}^{(2)} = (0, .443, .557)\begin{pmatrix} .416 \\ 1 \\ .727 \end{pmatrix} = .848. \tag{2.16}$$

Continuing this way each element of $\mathbf{P}^{(2)}$ and $\mathbf{Q}^{(2)}$ is found. The matrix for each successive time step $t+1$ can be constructed by taking columns from $\mathbf{P}^{(t)}$ and $\mathbf{Q}^{(t)}$ and using them to construct rows from the original one-step **P** and **Q** matrices. The complete results for the first five time steps and the asymptotic distribution interval are listed in Figure 2.7.

Time	Minimum (P) matrix			Maximum (Q) matrix		
	a	g	o	a	g	o
0	0.4363	0.0800	0.2976	0.5648	0.2342	0.4163
	0	0	0.5568	0.0813	0.4432	1
	0.0124	0.2415	0.4398	0.0987	0.5048	0.7270
1	0.1944	0.1093	0.3688	0.3605	0.3777	0.6554
	0.0069	0.1344	0.4282	0.1366	0.5048	0.8480
	0.0114	0.1106	0.4540	0.1405	0.4854	0.8610
2	0.0902	0.1118	0.4024	0.2645	0.4429	0.7694
	0.0094	0.1105	0.4377	0.1584	0.4940	0.8610
	0.0114	0.1162	0.4337	0.1613	0.4938	0.8550
3	0.0453	0.1128	0.4163	0.2193	0.4717	0.8191
	0.0105	0.1137	0.4311	0.1697	0.4939	0.8577
	0.0114	0.1131	0.4316	0.1708	0.4933	0.8570
4	0.0260	0.1130	0.4229	0.1981	0.4840	0.8406
	0.0110	0.1131	0.4302	0.1747	0.4935	0.8573
	0.0113	0.1132	0.4299	0.1753	0.4933	0.8568
5	0.0176	0.1131	0.4260	0.1881	0.4893	0.8499
	0.0112	0.1132	0.4293	0.1772	0.4934	0.8570
	0.0113	0.1131	0.4293	0.1774	0.4933	0.8569
∞	0.0113	0.1131	0.4287	0.1793	0.4933	0.8569
	0.0113	0.1131	0.4287	0.1793	0.4933	0.8569
	0.0113	0.1131	0.4287	0.1793	0.4933	0.8569

Figure 2.7: Matrices of parameter intervals calculated by the Hi-Lo method for times 0-5 and stable distribution ($t = \infty$) of a three-state grassland. Columns and rows represent a = annuals, g = perennial graminoids and o = other perennials, respectively.

Iterating by Monte Carlo simulation

A Markov set-chain can be simulated with a Monte Carlo approach. For each time step, a Markov chain matrix is created by drawing a random transition probability from within the interval (p_{ij}, q_{ij}) for each i and j. Next, one element of each row is chosen at random and adjusted so that the row sums to 1. At each time step, a new matrix is constructed, and the next-step transition probabilities calculated via matrix multiplication. This approach is computationally quick and simple. However, it is not deterministic. Starting with the same matrices and initial conditions yields a different result for each run. No stable asymptotic behavior results. By doing a large number of simulations, however, it is possible to get an idea of the asymptotic boundaries on the behavior.

For the two-state and three-state grassland models, we did simulations to estimate the boundaries on the asymptotic distribution of individuals among states over time. Transition probabilities were drawn from a uniform distribution. The system ran for 100 steps and we recorded the predicted frequency of each state at each step. Figure 2.8 shows the results of 1000 runs for the two state system at times up to 14 years, compared to the boundaries predicted by the Hi-Lo method. Even taking the minimum and maximum boundaries observed in 10,000 simulations, we do not get a perfectly accurate estimate of the boundaries for the Markov set-chain, although we are fairly close (Table 2.5).

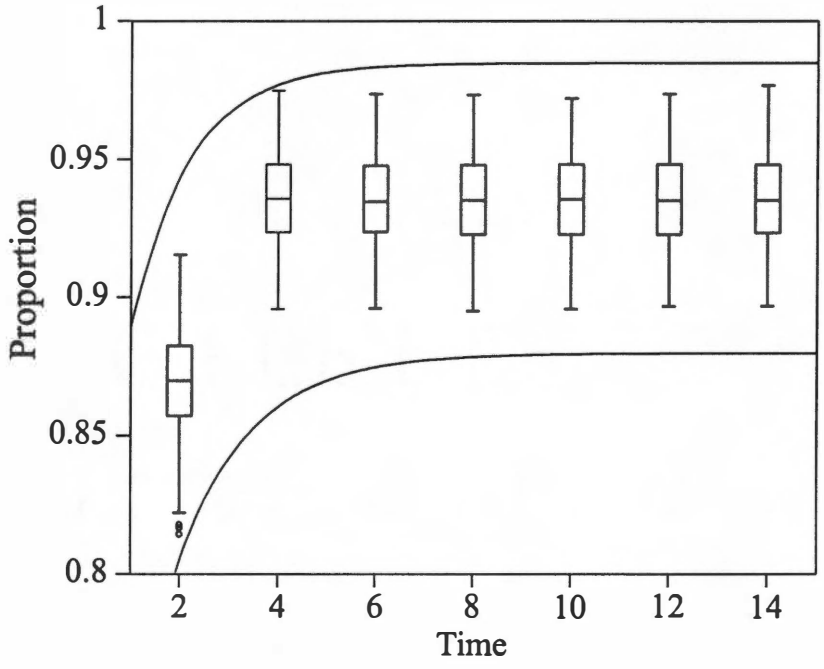


Figure 2.8: Box and whisker plot showing proportion of cover by perennials predicted in 1000 Monte Carlo simulation runs and interval boundaries predicted by the Hi-Lo method. Solid curves mark the Hi-Lo method boundaries. In the box and whisker diagram the 'whiskers' represent the upper and lower quartiles of the data. The whiskers end at the highest and lowest values observed over the 1000 runs, except for t=2, where low outliers are marked with open circles. Monte Carlo simulation predictions fall within tighter intervals than those predicted by the Hi-Lo method.

Table 2.5: Boundaries on distribution among states after 100 time steps for two-state and three-state system estimated by two methods. Monte Carlo boundaries are the minimum and maximum predictions from 10,000 runs.

	Hi-Lo		Monte Carlo	
Functional Group	**Min**	**Max**	**Min**	**Max**
Annual	0.0152	0.1203	0.0237	0.1081
Perennial	0.8797	0.9848	0.8919	0.9763
Annual	0.0113	0.1861	0.0346	0.1462
Perennial graminoid	0.1016	0.5624	0.1380	0.4578
Other perennial	0.3850	0.9768	0.4612	0.7878

Discussion

I have reported on a new extension of Markov chains that solves a major problem previously limiting their utility. Markov chains have been used to model community development. However, these models are problematic because of dynamic or uncertain transition probabilities. Markov set-chains with parameter intervals that encompass the breadth of the variation may be a viable approach to understanding and predicting succession. In addition to forecasting community composition, Markov set-chain models offer insight into the impact of variation, and how variation leads to uncertainty about community structure.

The relationship between iterating and simulating set-chains bears further exploration. How do the degree and effects of estimation compare between the Hi-Lo method and Monte Carlo simulation? What is the value of computing for exact boundaries when we can obtain a reasonable estimate by simulating? Also, solving the Markov set-chain gives us the boundaries on the distribution space, but does not tell anything about the relative frequency of different distributions of individuals among states within the bounds. Simulations assuming different types of distributions of parameters within the interval may help understand the importance of the boundaries. Are these minima and maxima unlikely absolutes, or frequently visited (or approached) borders?

Another set of questions is related to applying set-chains with real systems and data sets. The ability to deal with uncertain parameter values is a major selling point for set-chains, but how broad can the range of parameters and description of the system be? At what point is the parameter interval so large that the predictions tell nothing more than that each state will make up somewhere between 0 and 100 percent of the community? Also, in all the examples here, I have considered systems with two and three states. What happens when we consider bigger matrices? While the challenge with regular Markov chains was to pin down a single constant transition probability, set-chains have their own difficulty. Set-chains require that we know the highest and lowest possible value for each transition probability. How much information and what kind of data are needed in order to get a good estimate of these boundaries? These questions will be addressed in a future paper.

Predictions from Markov set-chains include a key feature that regular Markov chains do not. The breadth of the interval containing all possible distributions gives an idea of how precisely we can predict the role of each functional group in the community structure. Set-chains may also help examine how the predictability of a system changes over time. Early in development, there may be many possibilities and combinations of species coexisting, whereas later on in the process the system converges to a smaller range of states.

Markov set-chains may be particularly helpful as a tool for ecosystem restoration or management. The planning and implementation of restoration projects is fraught with uncertainty. Restoration is undertaken with a desired community or endpoint in mind, but little is understood about how planting different species combinations or ratios influences the result (Lockwood, 1997). A Markov set-chain could allow us to consider the effects of variation in the initial conditions and transition probabilities. This can tell about the relative importance of managing both the structure (composition) and function (process) during restoration. Set-chains may help us see how important it is to carefully select and control the initial ensemble of species, versus controlling the balance of competition that leads to directional replacement of species or functional groups.

CHAPTER 3

VARIABILITY THRESHOLDS IN ECOLOGICAL COMMUNITIES

In this paper I use a new modeling tool, Markov set-chains, to help understand patterns of variability in ecological community composition. Markov set-chains use intervals of possible transition probabilities. Set-chain predictions are also intervals, which give the ranges of possible species frequencies. The widths of the predicted distribution intervals are a measure of the variety of possible community structures that can arise. By working with intervals rather than mean values, Markov set-chains provide a unique window into the variability of community composition.

Variability in ecological communities has been studied for its causes and consequences (May 1973, Connell and Sousa 1983, Pimm 1984, 1991). More recently, however, variability has received attention as an attribute that can provide information about patterns of community dynamics (Micheli et al. 1999). Variability is a component of the multifaceted ecological concept, "stability" (Pimm 1991, Grimm and Wissel 1997). For this reason, variability is a factor in the long-standing discussion of the relationship between diversity and ecosystem function (see reviews in Pimm 1991, and Loreau 2001).

Community variability has been linked to a variety of abiotic and biotic factors (Micheli et al. 1999). A particularly great deal of interest has been focused on the influence of diversity on variability. Studies of the relationship between diversity and variability are divided among experimental and modeling approaches. Experimental

studies have found that increased diversity is associated with significantly lower variability in grasslands (Tilman and Downing 1994, Tilman 1996) and aquatic microcosms (McGrady-Steed et al 1997, Naeem and Li 1997). These findings are intriguing as they seem to contradict the long-prevailing theory that increasing species diversity, and thus the number and variety of interactions in a community, is destabilizing (May 1972, 1973).

Various theoretical and modeling approaches have been proposed to explain the mechanism underlying the experimental evidence that increased diversity leads to lower variability (Micheli et al. 1999, Loreau 2000). One idea is that within-functional group diversity serves as "biological insurance" (Walker 1992, Lawton and Brown 1993, Yachi and Loreau 1999). If one species is lost or perturbed, other functional group members can compensate, buffering the effect on the community as a whole. Probabilistic models have been used to demonstrate that ecosystems with more functionally redundant species have a greater chance of continued, reliable functioning (Naeem 1998, Rastetter et al. 1999, Nijs and Impens 2000).

Another explanation is that the apparent change in variability is a statistical by-product of classifying of community structure according into functional groups rather than considering individual species (Doak et al. 1998, Tilman 1998). This is based on the idea of Patten (1975), who showed that the aggregate density of many temporally fluctuating species appears less variable than the densities of the individual component species. This is due to an averaging effect that produces a smoothed, less variable view

of community dynamics. Increasing the number of species enhances the smoothing effect.

Models of competitive systems have been used to investigate the relationship between diversity and variability (Hughes and Roughgarden 1998, Yachi and Loreau 1999, Ives et al. 2000). These studies address a mechanistic question at the heart of the apparent contradiction between older models and new empirical results: How can having more or stronger interactions among species make a system less variable? Hughes and Roughgarden (1998) and Ives et al. (2000) used models of competitive systems to examine effects of species diversity on variation in biomass. Each concluded that the strength of interspecific competitive interactions has little effect on temporal variability. However, the degree of asymmetry of competitive interactions does appear to be an important factor (Hughes and Roughgarden 1998, Yachi and Loreau 1999). Additionally, Ives et al. (1999, 2000) showed that when species differ in their responses to environmental fluctuations, increasing diversity leads to decreased variability. These models all support the insurance hypothesis, showing that diversity among species buffers variability in biomass that can occur when species are lost. However, they emphasize a different aspect from the models of Naeem (1998) and Nijs and Impens (2000), as the buffering depends on differences between species and asymmetries in competition rather than redundancy.

The work I present here is the first I know of to combine an analytic model with data from a field experiment to examine factors influencing community variability. The set-chain model I use is probabilistic, not specifically allied with any of the mechanisms

outlined above. I study the community composition and rate of succession in artificially constructed grassland communities (Piper and Pimm 2001, in review). These experimental grassland plots were planted with four different species mixtures, varying in species composition and diversity. The species composition of all plots was measured annually and evaluated to see how quickly the systems approached a "prairie-like" target state. Piper and Pimm (2001, in review) concluded that more diverse plantings came closer to the desired composition, and did so more quickly. I reexamine their data from a functional-groups perspective and use a set-chain to model the systems. Markov set-chain models include in their predictions information about the range of species frequencies that may result, and thus how predictable the system is. Estimated set-chains for the more diverse plots predict narrower species ranges than do those for the less diverse plantings. I analyze a restricted Markov set-chain model to help understand what gives rise to this difference in ranges for different species mixtures.

Controlling the range of communities that may develop is an important goal of ecological restoration and management. In restoration, the bulk of the manipulation or management of a system most often takes place in the first one or two years of a project (Lockwood, 1997). However, the success of the project depends on fairly narrowly defined goals for that community far in the future. We know that communities that start out under similar conditions can diverge and become different in composition over time (Samuels and Drake 1997). A Markov set-chain model predicts how wide the range of possible future community configurations may be. Using set-chain models we can gain insight into what makes the difference between a community having a wide range of

possible outcomes, and one where composition is guaranteed to fall within a fairly narrow range. This insight into predictability could make set-chains a valuable tool for ecosystem management.

In the first section of this paper, I give an overview of the grassland experiment and explain how I processed the data to obtain time series for frequencies of annuals and perennials. Markov set-chain models make it possible to examine patterns of variability in these experimental communities. In the second section I explain how I set up and parameterized Markov set-chain models of the grassland experiment. The more diverse treatments lead to parameter intervals with greater values for the replacement probability of annuals by perennials. Set-chains with faster replacement of annuals by perennials give narrower species frequency ranges. Next, I formulate a restricted Markov set-chain to investigate this narrowing analytically. I show that there are variability thresholds, curves in parameter space, which define the conditions for variability to either increase or decrease over time. Understanding the effects of particular species and of attributes of species mixtures, such as diversity, on transition probability intervals may lead to prescriptive guidelines for making restored or constructed communities more predictable.

Field experiment data provides parameters for a set-chain model

A grassland restoration experiment was conducted at The Land Institute in central Kansas. Four replicate 32 m^2 plots on former agricultural land were seeded with perennial grassland species in four different species mixtures. Treatments varied in diversity (four, eight, twelve, and sixteen species), and composition was nested such that the eight species treatment included all species from the four species treatment, and so

on. Piper and Pimm (2001, in review) provide a detailed account of the experiment. Community composition was observed annually in July 1994-1999. Vegetation was sampled in twelve 0.75 x 0.75 *m*, square quadrats within each plot using cover class estimates by species (Daubenmire 1959). The data set provides a view of how species composition changed over time in plots where different species combinations were planted.

Piper and Pimm (2001, in review) focused on the influence of the different-diversity plantings on the rate at which a community approaches a predefined target state. The target was a "prairie-like" community with at least 100% foliar cover by native perennials. (100% foliar cover is distinct from 100% *of* cover, as each species' cover was estimated independently and layers of growth often overlap, creating totals greater than 100%.) Piper and Pimm concluded that plots planted with more species approached the target state faster.

At the most basic level, grassland succession may be viewed as a process of native perennials replacing weedy annuals (Holt et al. 1995). To keep things simple, all the analysis in this paper is done using a two-state representation of the system, where the community is made up of annuals and perennials. Starting with the species composition data collected by Piper and Pimm (2001, in review), I classified all species in the experiment as either annual or perennial. I then calculated the proportion of total cover that was annual and perennial each year, thus obtaining a six-year time series of frequencies of annuals and perennials in each plot (Table 3.1). Composition is expected to and does vary across plots of the same treatment due to environmental and other

Table 3.1: Relative proportions of annual and perennial plant cover over six years. An increase in the perennial cover indicates movement toward the target community. A = annuals, P = perennials.

# of species planted	Year	Replicate							
		1		2		3		4	
		A	P	A	P	A	P	A	P
4	1994	0.773	0.227	0.588	0.412	0.763	0.237	0.669	0.331
	1995	0.688	0.312	0.656	0.344	0.550	0.450	0.661	0.339
	1996	0.736	0.264	0.590	0.410	0.632	0.368	0.627	0.373
	1997	0.569	0.431	0.626	0.374	0.556	0.444	0.411	0.589
	1998	0.319	0.681	0.398	0.602	0.288	0.712	0.078	0.922
	1999	0.366	0.634	0.198	0.802	0.297	0.703	0.073	0.927
8	1994	0.715	0.285	0.792	0.208	0.640	0.360	0.677	0.323
	1995	0.539	0.461	0.636	0.364	0.589	0.411	0.557	0.443
	1996	0.538	0.462	0.499	0.501	0.438	0.562	0.457	0.543
	1997	0.382	0.618	0.131	0.869	0.249	0.751	0.075	0.925
	1998	0.187	0.813	0.060	0.940	0.236	0.764	0.077	0.923
	1999	0.120	0.880	0.125	0.875	0.195	0.805	0.054	0.946
12	1994	0.740	0.260	0.706	0.294	0.538	0.462	0.604	0.396
	1995	0.381	0.619	0.589	0.411	0.406	0.594	0.596	0.404
	1996	0.152	0.848	0.318	0.682	0.180	0.820	0.243	0.757
	1997	0.097	0.903	0.154	0.846	0.081	0.919	0.132	0.868
	1998	0.102	0.898	0.125	0.875	0.154	0.846	0.052	0.948
	1999	0.150	0.850	0.038	0.962	0.074	0.926	0.056	0.944
16	1994	0.692	0.308	0.682	0.318	0.567	0.433	0.522	0.478
	1995	0.315	0.685	0.461	0.539	0.327	0.673	0.358	0.642
	1996	0.226	0.774	0.309	0.691	0.263	0.737	0.119	0.881
	1997	0.075	0.925	0.037	0.963	0.063	0.937	0.106	0.894
	1998	0.020	0.980	0.013	0.987	0.029	0.971	0.062	0.938
	1999	0.070	0.930	0.033	0.967	0.127	0.873	0.134	0.866

chance differences (Figure 3.1, Figure 3.2). The time series provide information about community change over time that can be used to derive Markovian transition probabilities.

Markov set-chains

A Markov set-chain model is an extension of a regular Markov chain and shares many of its fundamental attributes. A Markov chain model represents succession as a probabilistic plant-by-plant replacement process. The system is made up of individuals in discrete states. (The grassland model has two states, annuals and perennials, which I label states 1 and 2.) A Markovian transition probability, p_{ij} is the chance that an individual in state i is replaced by one in state j over one time step (here, one year). A Markov chain model is often represented as a square matrix of all transition probabilities from every initial state i to every destination state j. The probability of transition between any pair of states in more than one step is the sum of the probabilities of making that transition by every possible path. For example, there are two ways for a perennial to replace an annual in two steps. It may stay annual the first year with probability p_{11}, and then be replaced by a perennial the second year with probability p_{12}. Alternatively, it may be replaced by a perennial the first year, with probability p_{12} and then remain perennial the second with probability p_{22}. The total two-year probability of transition from annual to perennial is thus $p_{11}\,p_{12} + p_{12}p_{22}$. Extended to any number of states, this process of adding probabilities is equivalent to matrix multiplication. Detailed introductory discussions of Markov chains with an emphasis on ecological applications are found in Guttorp (1995) and Jeffers (1978).

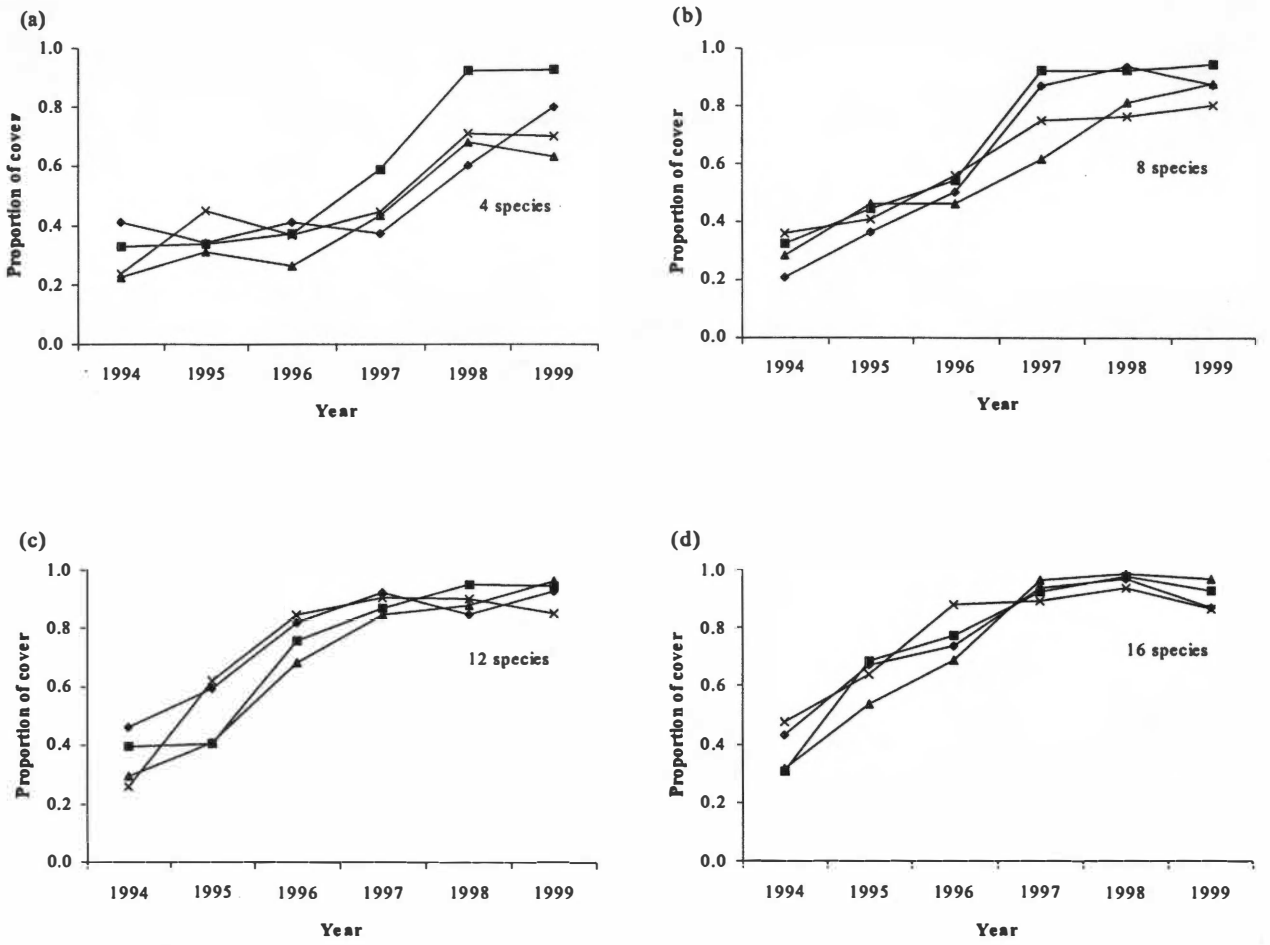


Figure 3.1: Relative proportion of perennial plant cover in four replicates of plots planted with four, eight, twelve and sixteen species.

(a)

Replicate		Treatment: 4 species		8 species		12 species		16 species	
		A	P	A	P	A	P	A	P
1	A	0.8252	0.1748	0.7578	0.2422	0.4386	0.5614	0.4422	0.5578
	P	0.0756	0.9244	0	1	0.0686	0.9314	0.0331	0.9669
2	A	0.8741	0.1259	0.6635	0.3365	0.6676	0.3324	0.5670	0.4330
	P	0	1	0.0328	0.9672	0	1	0.0086	0.9914
3	A	0.7170	0.2830	0.7918	0.2082	0.5425	0.4575	0.4824	0.5176
	P	0.1450	0.8550	0.0052	0.9948	0.0463	0.9537	0.0568	0.9432
4	A	0.7855	0.2145	0.6680	0.3320	0.6699	0.3301	0.4785	0.5215
	P	0	1	0.0064	0.9936	0.0016	0.9984	0.0592	0.9408

(b)

		A	P	A	P	A	P	A	P
Min	A	0.7170	0.1259	0.6635	0.2082	0.4386	0.3301	0.4422	0.4330
	P	0	0.8550	0	0.9672	0	0.9314	0.0086	0.9408
Max	A	0.8741	0.2830	0.7918	0.3365	0.6699	0.5614	0.5670	0.5578
	P	0.1450	1	0.0328	1	0.0686	1	0.0592	0.9914

Figure 3.2: Markov chains and set-chains for four diversity treatments. (a) Markov chains for replicates of the same treatment vary. (b) The set-chain minimum and maximum matrices contain the highest and lowest observed value for each parameter, respectively. The width of the interval between the minimum and maximum for each parameter is relatively consistent across treatments. However, the mean value changes considerably; treatments where more species of perennials were introduced have a higher probability of annuals being replaced by perennials. A = annuals, P = perennials.

Each time series in the grassland experiment may be simulated as a Markovian sequence of functional group replacements. We know the frequencies of each group at each time, but not specifically what replacements generated the changes. To estimate the transition probabilities, I used a least squares estimation method (Lee et al. 1977) that finds the Markov chain that best describes a time series. To estimate, I used a Fortran program for a quadratic programming algorithm obtained from Lee's (2000) website. Four different Markov chains describe the four time series for each set of replicate plots (Figure 3.2a). When iterated, each Markov chain predicts a different unique series of community compositions (Figure 3.3). The four Markov chains define a range of values for each transition probability. This range designates the interval for a Markov set-chain (Figure 3.2b). The composition at the start of the experiment in 1994 (Figure 3.1) was different in each plot. This range defines an initial conditions interval for each functional group among plots planted with the same species mixture.

The existence of a range of different Markov chains for the same treatment precludes the use of a regular Markov chain. Markov set-chains were developed to accommodate precisely this situation. Hartfiel (1998, 1991) has introduced set-chain models into the mathematical literature. Samuels and Kot (2001, in review) discuss methods for iterating set-chains to find the distribution of individuals among states over time and asymptotically, in the limit of large time. They also highlight the potential of set-chains as models of succession. A Markov set-chain uses ranges of possible transition probabilities instead of constant transition probabilities. For example, in the sixteen-species system (Figure 3.2), the probability, p_{11}, of an annual remaining an annual

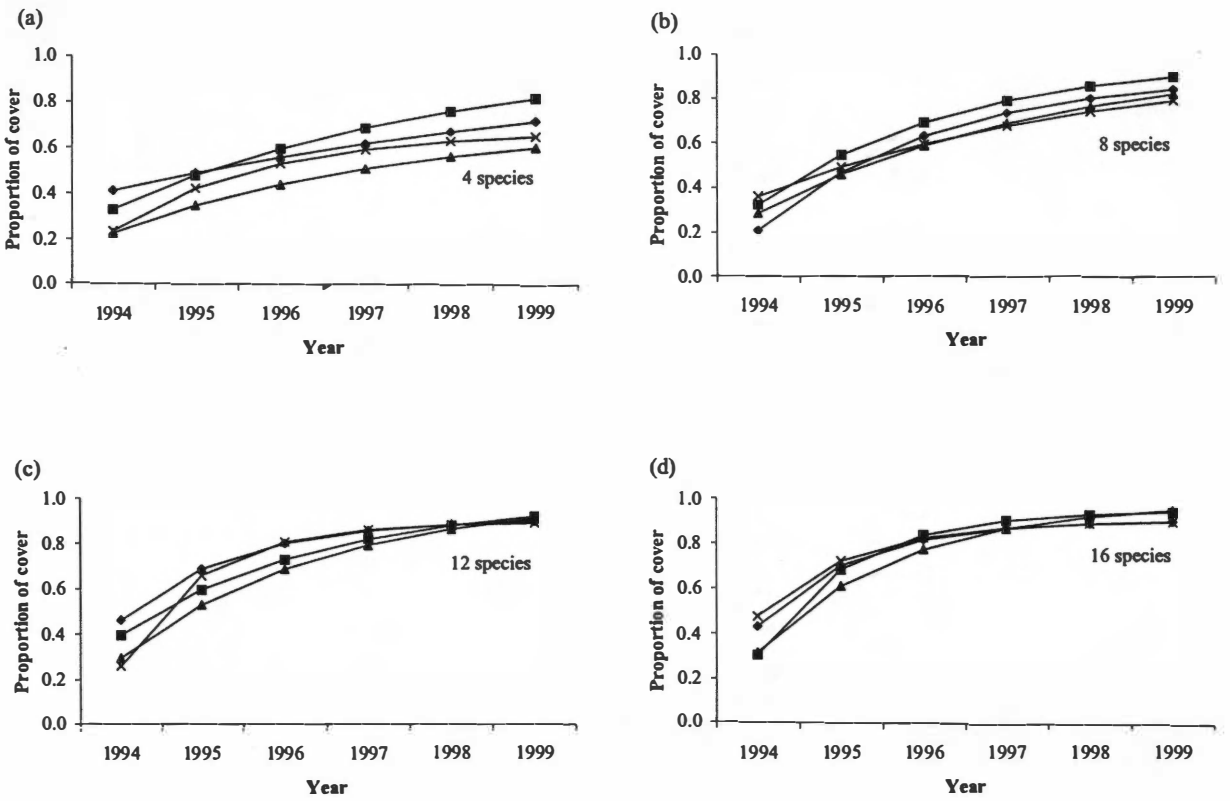


Figure 3.3: Proportion of perennial plant cover predicted by regular Markov chains for four replicate plots planted with four, eight, twelve and sixteen species.

after one year, has four different estimated values ranging from 0.4422 to 0.5670. To formulate a Markov set-chain I took the minimum and maximum estimated values for each transition probability as the boundaries of an interval of transition probabilities. The interval defines the range of all possible values that are likely to occur for that particular transition probability. In any given year, each transition probability must fall within its range.

Given the transition probability intervals and initial condition intervals, it is possible to find unique boundaries on the distribution at each future time and asymptotically. Methods for iterating and finding asymptotic distributions for any size system are discussed in detail in Samuels and Kot (2001, in review) and Hartfiel (1998). The calculations for iterating a two-state set-chain are demonstrated in the next section. Using the set-chain and the range of observed initial conditions (Table 3.1, 1994 frequencies) for this two-state system, I iterated each set-chain to predict the boundaries on the distribution in future years (Figure 3.4). In this case, because there are only two functional groups, the interval widths of frequencies of annuals and perennials are the same. When intervals of frequencies widen or narrow over time for one functional group, they also do so for the other.

The transition probabilities for annuals strongly affect the behavior of the iterated set-chains. Each interval in Figure 3.4 gives all possible frequencies of perennials at one specific time. The width of an interval is a measure of the variability in community composition. The asymptotic width for perennials decreases as the rates of replacement of annuals by perennials increase. The change in species frequency interval width over

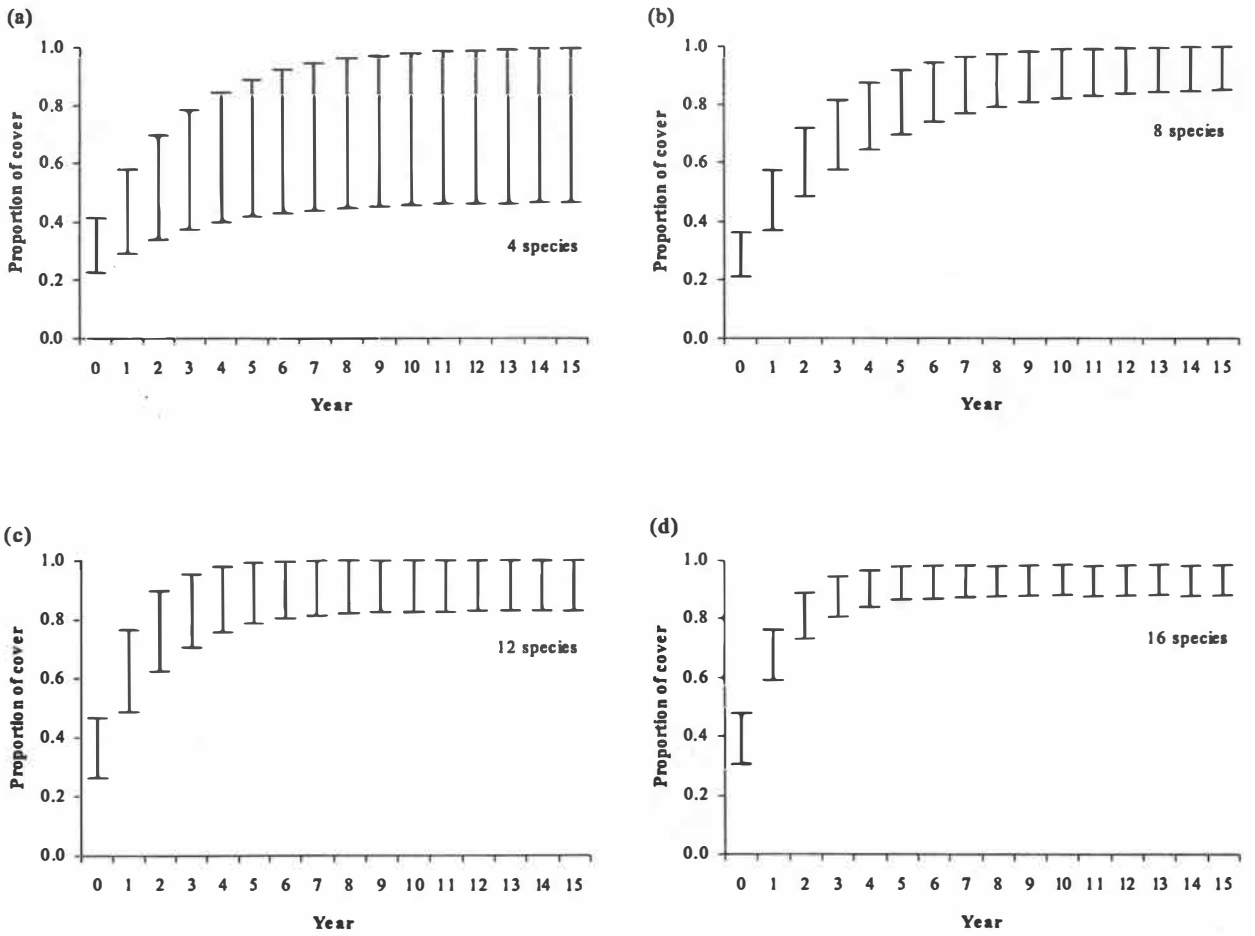


Figure 3.4: Predicted intervals of perennial cover over fifteen years for plots planted with (a) four, (b) eight, (c) twelve, and (d) sixteen species. Annual and perennial cover always sum to 1, so interval widths for annuals are the same as for perennials. Intervals for plots planted with four species are all wider than the initial interval. Intervals for eight- and twelve-species plots widen initially, but asymptotically are narrower than the initial interval. Intervals for the sixteen-species plots become progressively narrower.

time is also quite interesting. For the treatment with four planted species, the perennial frequency interval widens each year, up to the asymptotic width. On the other hand, for the treatment with sixteen planted species, the interval of possible frequencies for perennials narrows over time. The frequency intervals of the eight- and twelve-species treatments widen during the first year, but then narrow to their asymptotic width. Additionally, the more diverse plots converge faster than less diverse plantings. Notably, the minimum frequencies do not necessarily converge at the same rate as the maximum frequencies.

Variability thresholds and community predictability

Markov set-chain models predict that plots with higher probabilities of replacement of annuals by perennials not only achieve the desired state more quickly, but also have a narrower range of possible functional group frequencies than those with lower replacement probabilities. I wanted to understand what controls the range of frequencies of annuals and perennials. I began by examining the parameter values in the set-chains to formulate an idea about what drives the differences between plots where different species combinations were planted. I then constructed and analyzed a restricted Markov set-chain to investigate the relationship between transition probability intervals and species frequency ranges.

Comparing the annual-to-perennial transition probabilities in the first row of each set-chain matrix (Figure 3.2b), the plots planted with more species of native perennials have faster rates of replacement of annuals. The mean probability of an annual being replaced by a perennial, a_{12}, increases progressively from .20 with four species to .51

with sixteen species. In contrast, the perennial-to-annual transition probabilities are relatively invariant, with a_{21} ranging from .01 to .05. Somehow, adding more kinds of perennials changes the fate of the annuals, but does not substantially affect whether perennials remain once they arrive. I take advantage of the relative constancy of rates of replacement of perennials in formulating the restricted set-chain below.

A restricted set-chain for grassland succession

Both the time series from the grassland experiment and my set-chain simulations suggest that the systems planted with more diverse species combinations vary less in functional-groups composition over time. To try and unravel why this occurs, I analyze a restricted two-state Markov set-chain. I begin with the set-chain

$$\mathbf{P} = \begin{bmatrix} a_{11} & 1 - A_{11} \\ c & 1 - c \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} A_{11} & 1 - a_{11} \\ c & 1 - c \end{bmatrix}. \tag{3.1}$$

The first element of each row and column is a transition probability for annuals. The second element represents perennials. **P** and **Q** contain all the minimum and maximum possible one-step transition probabilities, respectively. For example, a_{11} is the minimum and A_{11} the maximum chance of an annual remaining annual after one year. The probability of a perennial being replaced by an annual is taken as constant, c. We also specify an initial conditions interval

$$(x_0, 1 - X_0), (X_0, 1 - x_0). \tag{3.2}$$

X_0 is the maximum initial proportion of cover by annuals; x_0 is the minimum. The initial frequency of perennials ranges from $1 - X_0$ to $1 - x_0$.

Predicting the range of frequencies

Each transition probability interval gives the one-step chance of making a particular transition. The probability of making a transition in more than one step is the sum of the probabilities of all possible paths between the two specified states in the given number of steps. Markov set-chain predictions for each future step involve minimizing (or maximizing) the multi-step transition probability. How this is done depends on the relative values of c, a_{11} and A_{11}. Taking into account that $a_{11} \leq A_{11}$, there are three possible cases,

$$c \leq a_{11} \leq A_{11} \tag{3.3a}$$

$$a_{11} \leq c \leq A_{11} \tag{3.3b}$$

or

$$a_{11} \leq A_{11} \leq c. \tag{3.3c}$$

All of the set-chains in the grassland data have parameters that fit case 3.3a; the probability of a perennial being replaced by an annual is less than the minimum probability of an annual remaining annual in one year. So, in this section I present an analysis for case 3.3a. The results for 3.3b and 3.3c may be found in a similar manner and are summarized at the end of this section.

The initial conditions intervals 3.2 give the range of possible values for the frequency of each functional group at the beginning of the first year. The transition probability intervals 3.1 contain all the one-step probabilities of beginning in some state i and ending up in state j. The proportion of total cover in either state after more than one time step is found by minimizing or maximizing the sum of the probabilities of all

possible paths from a specified starting point i to end point j. There are, for example, two ways to make the two-step transition from annual to perennial: stay annual the first year and be replaced by a perennial the second, or be replaced by a perennial the first year and then stay perennial the second.

To find the minimum (or maximum) boundaries on this probability, I consider the scenario where the greatest possible proportion of individuals start out in the state that is least (or most, when maximizing) likely to lead to the specified endpoint. Here I focus on annuals, but the total of the proportion of annuals and perennials is 1, so the species frequency interval width is the same for both annuals and perennials. The first column of **P** gives the minimum chance of arriving at annuals from either state. Because a_{11} is greater than c, we choose the row vector (starting condition) that sets the largest possible proportion of the initial community as annuals. This way, the greatest possible part of the community will be subject to the smallest probability of transition to annuals. The minimum and maximum frequencies of annuals after one year, x_1 and X_1, are

$$x_1 = (x_0,\ 1 - x_0)\begin{bmatrix} a_{11} \\ c \end{bmatrix} = x_0 a_{11} + c\ (1 - x_0) \tag{3.4}$$

and

$$X_1 = (X_0,\ 1 - X_0)\begin{bmatrix} A_{11} \\ c \end{bmatrix} = X_0 A_{11} + c\ (1 - X_0). \tag{3.5}$$

The minimum, x^*, of the asymptotic interval of the proportion of annuals is defined as that value for which $x_1 = x_0$. Similarly, the maximum, X^*, of the interval

containing the proportion of annuals is that value for which $X_1 = X_0$. I replace x_0 and x_1 with x^*, and X_0 and X_1 with X^*. Then, solving for x^* and X^* gives

$$x^* = \frac{c}{1 - a_{11} + c} \tag{3.6}$$

and

$$X^* = \frac{c}{1 - A_{11} + c}. \tag{3.7}$$

Next, I use the expressions in 3.6 and 3.7 to find out when the initial interval is wider or narrower than the asymptotic interval. To find the condition on A_{11} such that the asymptotic interval of annuals is identical in width to the initial interval, I set

$$\begin{aligned} X^* - x^* &= X_0 - x_0 \\ \frac{c}{1 - A_{11} + c} - \frac{c}{1 - a_{11} + c} &= X_0 - x_0. \end{aligned} \tag{3.8}$$

Solving for A_{11} gives

$$A_{11} = \frac{a_{11}c + (X_0 - x_0)(1 + c)(1 + c - a_{11})}{c + (X_0 - x_0)(1 + c - a_{11})}. \tag{3.9}$$

This curve is shown in Figure 3.5. This curve divides the (a_{11}, A_{11}) parameter plane into two regions. Below the curve, the asymptotic interval is narrower than the initial interval, and above the curve it is wider. The points defined by the parameters a_{11} and A_{11} of the Markov set-chain models of the grassland experiment are also shown. Except for the plots planted with four species, all fall below the curve, indicating that the asymptotic interval is narrower than the initial range.

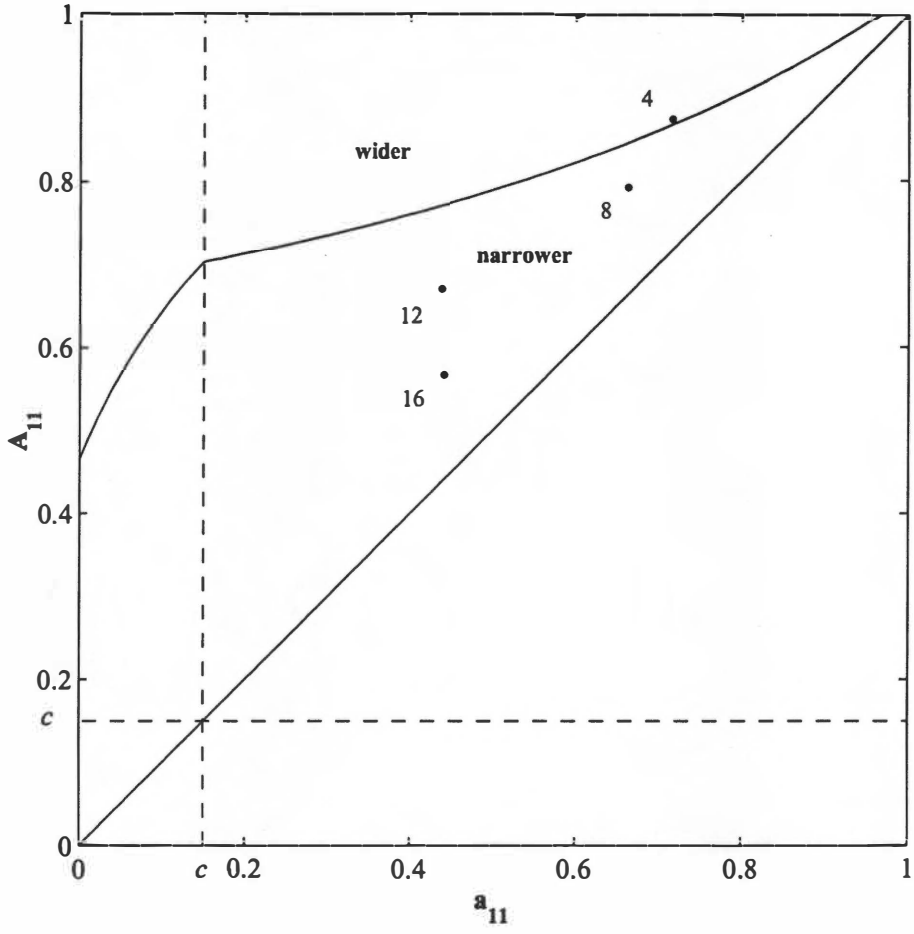


Figure 3.5: The curve that defines where the asymptotic interval is wider or narrower than the initial interval. Annual-to-annual transition probability interval parameter values for treatment plots planted with four, eight, twelve and sixteen species are marked. The interval for the four-species plots is wider asymptotically than the initial conditions interval, while the others are narrower. The curves are plotted for $c = .15$, $x = .588$, and $X = .773$.

Transient behavior

I have previously noted that, for the eight- and twelve- species treatment plots, the estimated set-chains predict that the annual frequency intervals will become wider after one year. However, the asymptotic interval is narrower than the initial interval. This type of transient amplification of perturbations has been noted and discussed in other contexts (Neubert and Caswell 1997, Jonsson and Trefethen 1998). Neubert and Caswell (1997) point out that many ecological studies focus on resilience, yet perturbations may initially grow, even when systems ultimately recover. When the aim is management of a system for a specific outcome, as in restoration, short-term transients may be particularly important.

Does the width of the interval containing possible community compositions widen or narrow during the first year of the experiment? I evaluate to find where the one-step interval width is the same as the initial interval. After setting

$$X_1 - x_1 = X_0 - x_0 \tag{3.10}$$

and inserting formulas for X_1 and x_1 from 3.4 and 3.5, we find that

$$X_0 A_{11} + c\,(1 - X_0) - x_0 a_{11} - c\,(1 - x_0) = X_0 - x_0 . \tag{3.11}$$

Solving for A_{11} now gives

$$A_{11} = \frac{(c+1)(X_0 - x_0) + x_0 a_{11}}{X_0} . \tag{3.12}$$

This new curve is shown in Figure 3.6. The regions are analogous to those in Figure 3.5. Below the curve, the first-step interval is narrower than the initial interval, and above the curve it is wider. The (a_{11}, A_{11}) points for the four grassland set-chains are also shown. With the exception of the sixteen-species plantings, all of the points are above the curve, indicating that the one-year interval of frequencies is wider than the initial interval.

Variability thresholds define four regions of parameter-space

Superimposing the two curves, we see that the (a_{11}, A_{11}) parameter plane is divided into four regions (Figure 3.7). The four regions represent all possible combinations of intervals widening or narrowing, initially or asymptotically. The parameters for each treatment fall neatly within the region that describes its dynamics noted in Figures 3.1 and 3.4. The predicted frequency intervals in plots with four planted species are always wider than the initial interval. The intervals for plantings of eight and twelve species widen initially, but are asymptotically narrower than the initial intervals. The plots planted with sixteen species have a predicted distribution interval that is consistently narrower than the initial interval. Notably, there is one region defined by the curves in Figure 3.7 that is not represented by any of the experimental plots. This region is the one marked IV, where the interval is initially narrower, but asymptotically wider, than the initial interval. Given the initial conditions and value of c from the grassland data, this behavior would require extremely high values for both the minimum and maximum probability of annuals remaining annuals. In all our treatments the annuals are replaced more frequently than would lead to this behavior.

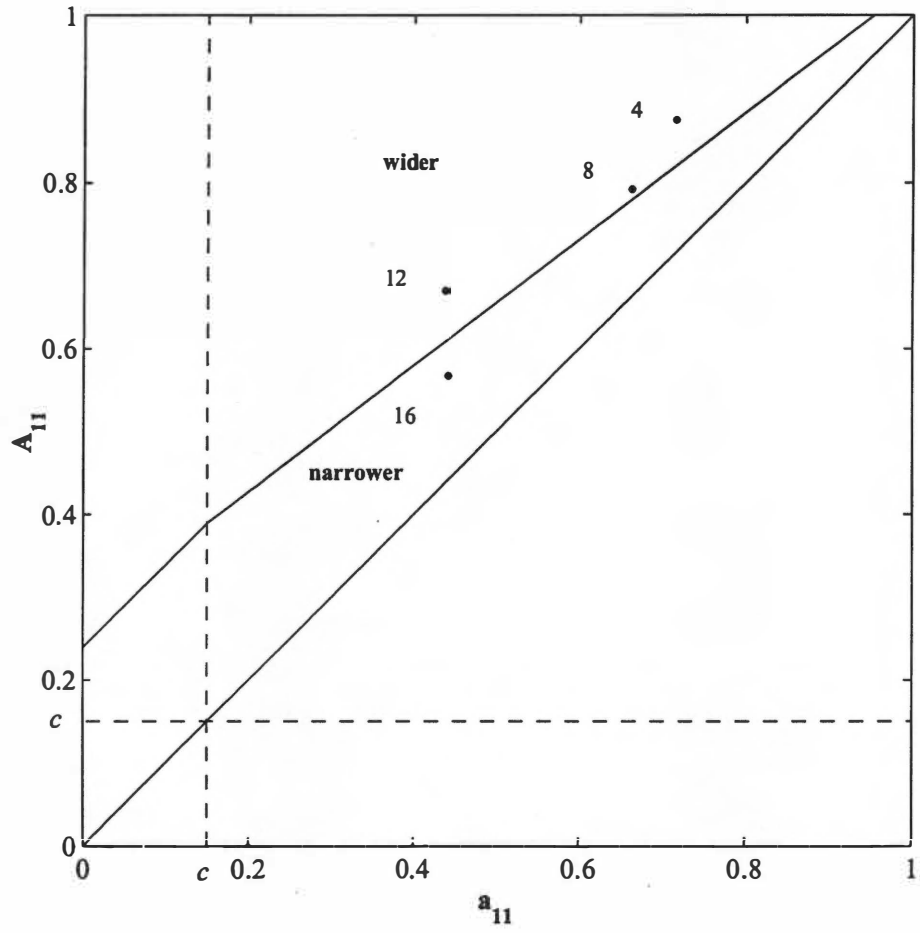


Figure 3.6: The curve that defines where the one-step interval is wider or narrower than the initial interval. Annual-to-annual transition probability interval parameter values for treatment plots planted with four, eight, twelve and sixteen species are marked. The interval for the sixteen-species plots after one year is narrower than the initial interval, while the intervals for the other plots are narrower. The curves are plotted for $c = .15$, $x = .588$, and $X = .773$.

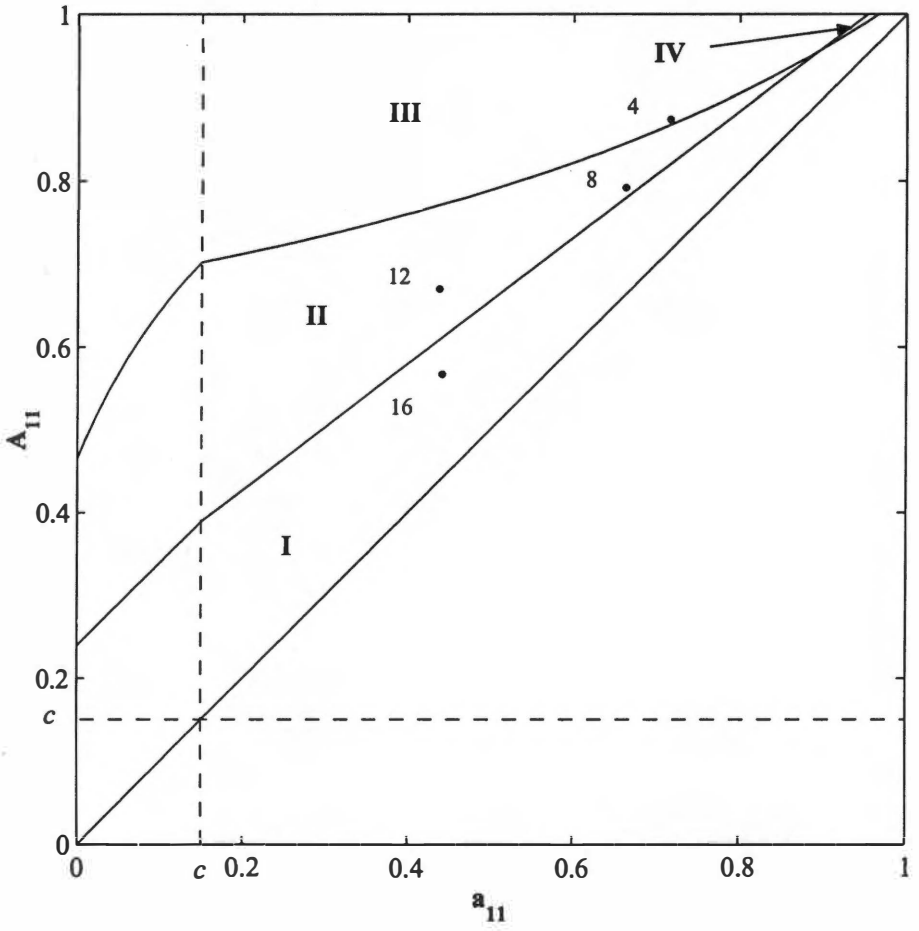


Figure 3.7: Four regions of transient and asymptotic interval size change. I. Interval is always narrower than the initial interval. II. One-step interval is wider, but asymptotic interval is narrower. III. Interval is always wider than the initial interval. IV. One-step interval is narrower, but asymptotic interval is wider. Annual-to-annual transition probability interval parameter values are marked for treatment plots planted with four, eight, twelve, and sixteen species. The curves are plotted for $c = .15$, $x = .588$, and $X = .773$.

Asymptotic and next-step interval widths for cases 3.3b and 3.3c

Curves for equal interval width for systems with parameters where $a_{11} \leq c \leq A_{11}$ (3.3b) and $a_{11} \leq A_{11} \leq c$ (3.3c) can be calculated in a similar manner as was done for the case where $c \leq a_{11} \leq A_{11}$ (3.3a) above. Asymptotic interval width equal to initial interval width occurs where, for case 3.3b,

$$A_{11} = \frac{(1-X_0)c^2 + (X_0 a_{11} - x_0)c + X_0(1+a_{11}) - x_0}{(1-X_0)c + X_0(1+a_{11}) - x_0}, \tag{3.13}$$

and, for case 3.3c,

$$A_{11} = (a_{11} - c + 1)\frac{X_0}{x_0} + c - 1. \tag{3.14}$$

The first-step (transient) interval is equal to initial interval width, where, for case 3.3b,

$$A_{11} = a_{11} + 1 - \frac{X_0}{x_0}, \tag{3.15}$$

and, for case 3.3c,

$$A_{11} = \frac{(1-c)(X_0 - x_0) + X_0 a_{11}}{x_0}. \tag{3.16}$$

Discussion

I have used a Markov set-chain to consider how transition probabilities influence variability of functional group composition in a grassland community. For the grassland experiment system, set-chains show the effect of the experimental treatments on variability. Set-chains predict the range of communities that may arise during or as a result of succession. Analysis of a restricted two-state model leads to a way of

predicting, for any initial conditions interval, when the dynamics of the developing community will become more or less variable over time. This view of the breadth of the range of outcomes of succession may be valuable for managers of restoration projects. Identifying what species combinations have transition probabilities that lead to increasingly certain outcomes will be of great value.

Changing the value of *c*, the probability that an annual replaces a perennial, can help us understand the importance of what happens to the annuals during succession. The regions depicted in Figures 3.5-3.7 were all modeled using the same parameters taken from the experimental data. In this scenario, *c* is .15. The starting proportion of annuals is between .588 and .773. Other parameters, however, can give rise to different situations. The predictability of the experimental set-chain depends on the guarantee that perennials remain established after they arrive. For example, consider a scenario where nearly all perennials remain perennial ($c = .01$) and the initial frequency of annuals is between .8 and .9 (Figure 3.8). In this case, all the experimental treatments have intervals that are wider after one step, but narrower asymptotically, than the initial interval.

Changing *c* so that perennials have only a .5 chance of remaining once they arrive leads to different results (Figure 3.9). Now the intervals for all treatments except for the sixteen-species widen, both after one year and asymptotically. The interval for the sixteen species treatment plots widens in the first year, but narrows asymptotically. It appears that the less certain the fate of the perennials, the more species it takes to ensure decreasing asymptotic interval width. In this case, the diversity 'cost' of predictability grows as the rate of change in the system increases.

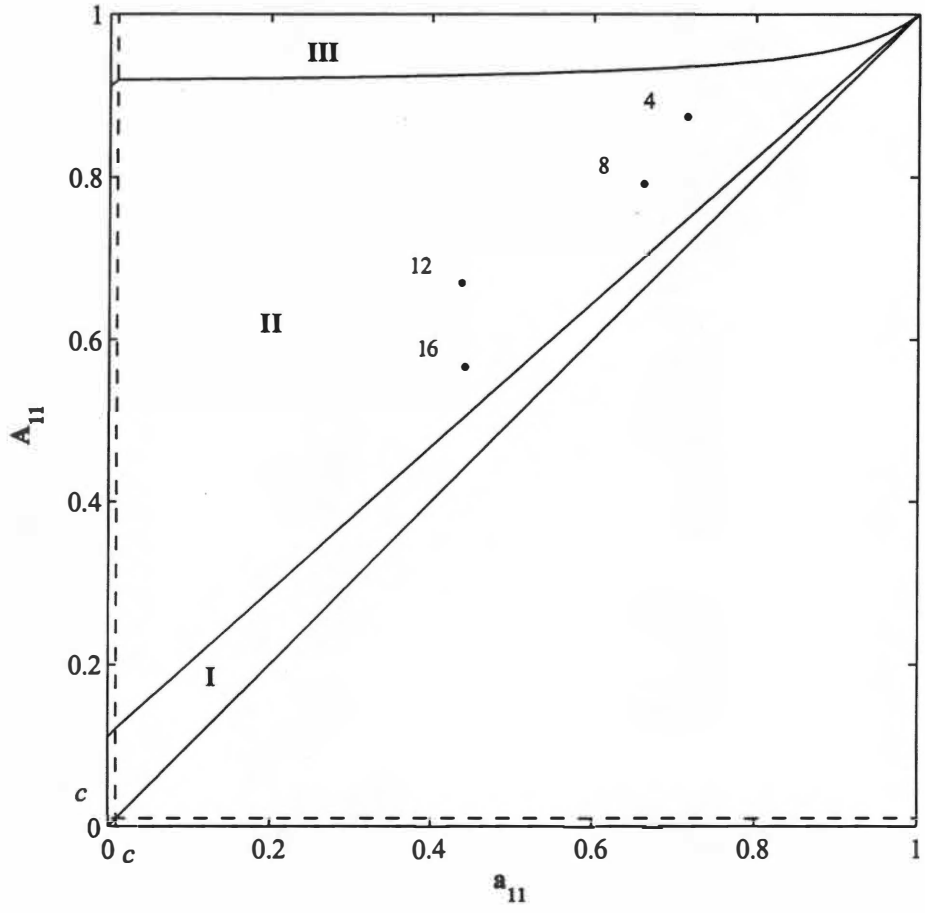


Figure 3.8: Four regions of transient and asymptotic interval size change for a system where perennials nearly always stay perennial (c = .01). I. Interval is always narrower than the initial interval. II. One-step interval is wider, but asymptotic interval is narrower. III. Interval is always wider than the initial interval. IV. One-step interval is narrower, but asymptotic interval is wider. Annual-to-annual transition probability interval parameter values are marked for treatment plots planted with four, eight, twelve, and sixteen species. The curves are plotted for x = .8, and X = .9. All the grassland experimental plot intervals are wider after one step, but narrower asymptotically.

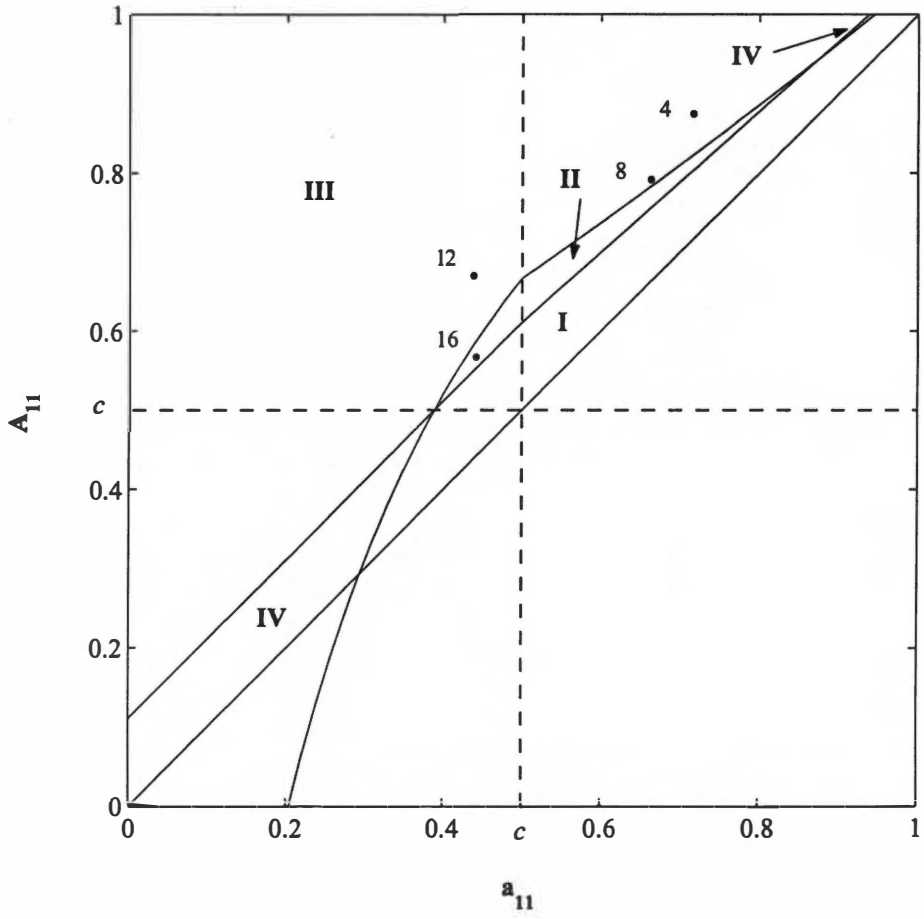


Figure 3.9: Four regions of transient and asymptotic interval size change for a system where perennials may be replaced by annuals half the time (c = .5). I. Interval is always narrower than the initial interval. II. One-step interval is wider, but asymptotic interval is narrower. III. Interval is always wider than the initial interval. IV. One-step interval is narrower, but asymptotic interval is wider. Annual-to-annual transition probability intervals are marked for treatment plots planted with four, eight, twelve and sixteen species. The curves are plotted for x = .9, and X = .9. Except for the sixteen-species plots, the intervals for all the grassland treatments are wider after one step and asymptotically.

This demonstration of the importance of the perennial transition probability c points out that what happens to the perennials can also be important in influencing system variability. In the analysis presented here, I have focused on the influence of the range of replacement probabilities for annuals. By holding perennial transition probabilities constant I have not explored the possible contribution of the perennial transition probability intervals to the system dynamics. I did perform an analysis of a restricted set-chain model where perennial transition probabilities are fluctuating and annuals are held constant. In this case, I get the same curves and regions as were obtained the analysis with fluctuating annuals (Figure 3.7), but in (p_{22}, P_{22}) parameter space instead of (p_{11}, P_{11}). However, the perennial-to-perennial transition probability intervals from the grassland the experiment do not fall in regions of parameter-space that explain the changes in variability predicted by the set-chain. This reinforces the idea that in the experimental systems, the change in annual transition probabilities is influencing the differences in variability.

In Markov set-chains, the asymptotic interval is typically independent of initial conditions. This is seen in equation (3.6) where x^* and X^* depend on c, a_{11} and A_{11}. However, all the variability thresholds here have been assessed relative to the width of the initial conditions interval. Essentially, we are allowing the initial conditions interval to set the benchmark for variability. For a wide initial interval width, in order for the predicted interval to be even wider requires extremely broad parameter intervals. Taking the initial interval width as the standard against which future predictions are measured makes sense for my grassland experiment example. We are taking the natural variability

measured at the start of the experiment as the standard for variation in the future. However, there are cases where this might not be the best approach. In a situation where we want to ensure that a system varies within some specified desired range, we might instead specify some desired interval width. This could be done by, in solving equation 3.9, replacing the initial conditions interval ($X_0 - x_0$) with some set width.

The relationship between variability and predictability is important to how we interpret the meaning of the variability thresholds. The predicted distribution interval is the range of possible states at any time. The set-chain parameter intervals give the range of possible Markov chains that might be acting in any given year. So, wider predicted distribution intervals mean that more different species compositions might occur in any given year. This does not necessarily mean that the composition will fluctuate more from year to year. If we knew the actual composition at some transient point, the ranges of possible species frequencies for the next year would be narrower than those predicted from some earlier initial conditions. Actual year-to-year variability is narrower than the interval width predicted by the set-chain. Of course, any time we know the composition exactly, then the initial interval size is zero. In that case, as long as the parameter intervals are not constant, the predicted distribution interval, one year or asymptotic, is always wider than the initial interval. In this situation it would be more useful to specify a standard for interval width and examine under what conditions the community is more or less variable than that value.

The variability thresholds explained in this paper were found by analyzing an extremely simple set-chain. Certainly, this raises the question of what other interesting types of behavior might arise in a less simplified set-chain. One next step might be to analyze a set-chain with intervals for both annual and perennial transition probabilities, instead of keeping perennials constant. Another possibility is to analyze a system with more states. However, adding even one state increases the number of parameters and degrees of freedom very quickly. These more complex set-chains could help provide a better view of the scope of dynamics that can arise from Markov set-chains. For example, Markov chains can give rise to oscillations. Do oscillations also occur in set-chains? In my simulations I found a variety of scenarios, but I have yet to find instances where the intervals grow, then shrink, and then grow again (oscillations). However, this does not necessarily mean oscillations cannot happen. We have looked at first year and asymptotic intervals relative to initial interval widths. However, this view does not actually delineate whether the interval becomes wider or narrower in each year-to-year step. The duration of the transient behavior, the amount of time spent with the intervals widening before they narrow, does decrease as the parameters approach the variability thresholds.

The specific species introduced, or aggregate factors like diversity, may influence transition probabilities and in turn variability. Identifying how these factors affect transition probability intervals may be a key to using set-chains as models to help guide ecological restoration or management. Introducing more species may make the critical difference between creating a community with a broad range of possible compositions

and one with a fairly precisely known outcome. Markov set-chains help us to understand dimensions of the relationship between diversity and ecosystem function that we previously could not address with models. This work demonstrates a unique and potentially useful application for set-chains. This approach may help us to understand the community dynamics in other data sets and plan more successful community management and restoration projects.

CHAPTER 4

ISSUES IN IMPLEMENTING MARKOV SET-CHAINS

Markov set-chains are a new approach for modeling succession (Samuels and Kot, 2001, in review). Developed by Hartfiel (1991, 1998), set-chains are an extension of regular Markov chains. Regular Markov chains model a system as a collection of individuals in some finite number of discrete states. Change occurs as a series of probabilistically driven individual-level replacements in discrete time. A problem that has limited the application of Markov chain models in ecology is that the transition probabilities must be precisely specified, and constant for all time. In natural systems, transition probabilities often fluctuate rather than maintain constant values. Markov set-chains solve this problem by replacing transition probabilities with intervals containing all possible values for the probability of each type of transition occurring. The result is a prediction about the boundaries on the interval containing all possible community structures that may occur at any future time and asymptotically. Set-chains are particularly promising because they retain many of the attributes that make Markov chain models appealing. A set-chain can be iterated to find the distribution interval at any time and, under most conditions, has a unique asymptotic distribution interval, independent of initial conditions.

The interval approach of Markov set-chains has pros and cons. Markov set-chains give a range of possibilities, rather than a single precise prediction. Additionally,

set-chain predictions are limited in that the *only* information they provide is the boundaries. The distribution of values within the predicted interval remains unknown. Yet what is lost in precision is gained in new information about variability, an aspect of community dynamics that regular Markov chains do not address. Iterated and asymptotic predictions from a Markov set-chain give the boundaries on the range of possible community structures. The width of the intervals containing possible frequencies of each species or functional group give a measure of the variability of system dynamics. Wider intervals mean there is a broader range of possible community structures. The predictions from a set-chain are also powerful in that they are quite robust. As long as the transition probability intervals are specified correctly, the predicted boundaries on the community composition are absolute limits on the possible behavior of the system. Because they offer insight into variability and limits on community dynamics, set-chains could be a powerful tool for ecosystem management. They can be used to project best- and worst-case scenarios, even without precise parameter measurements.

Markov set-chains solve the problem of fluctuating or imprecisely known transition probabilities, a key problem that has limited the application of Markov chains in ecology. While handling parameter uncertainty, however, set-chains do bring along an array of new questions. Set-chains allow for fluctuating parameters, but is there a limit to how much variability they can accommodate? Additionally, methods for iterating and finding asymptotic behavior of set-chains involve estimation. How good are the estimates? Are the predicted boundaries dependable enough to use in making management decisions? A further caveat is that, like most models, the parameterization

to represent a natural system depends on data, which is nearly always limited. What type of data, and how much, is needed to construct an accurate set-chain model?

The purpose of this paper is to examine those questions and their consequences for using set-chain models to study the dynamics of natural ecological systems. First, I give a brief introduction to Markov set-chains and their relationship to regular Markov chains. Next, I introduce the grassland example used in this paper and discuss how set-chains were derived from the data. The rest of the paper is then organized into sections to address three sources of error or instability in set-chain parameters: (1) error in measurement or due to natural fluctuation that affects the breadth of parameter intervals and the width of predicted distribution intervals, (2) inaccuracy that results from estimation in the algorithms for iterating and computing asymptotic behavior of Markov set-chains, (3) error in transition probability intervals derived from limited data due to short time series or small number of replicates. Each of these factors impacts the boundaries of the predicted distribution, the frequency of particular community states within the boundaries, and the range of possible successional pathways. I investigate each of the three factors, using simulations to learn more about its influence; for each, I discuss its implications for modeling and illustrate with examples from a grassland system. Understanding more about the mechanics of Markov set-chains will clarify our understanding of what they can and cannot do and how they may be applied to help understand the dynamics of ecological communities.

Markov chains and set-chains

Markov set-chains are an extension of regular Markov chains. The system being modeled is made up of individuals, x, where x is one particular individual. At any time each individual is in one of some finite number n of discrete states. In a regular Markov chain the community composition is given by the vector $\mathbf{b}_t = [b_{1,t}, b_{2,t}, \ldots, b_{i,t}, \ldots, b_{n,t}]$, where $b_{i,t}$ is the proportion of individuals in state i at time t. An individual in state i is replaced by one in state j with some characteristic transition probability,

$$a_{ij} = \Pr(x_{t+1} = j \mid x_t = i). \tag{4.1}$$

The proportion of individuals within the community in state i after one time-step is

$$b_{j,t+1} = \sum_{i=1}^{n} b_{i,t} a_{ij} \tag{4.2}$$

The transition matrix of a regular Markov chain,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \tag{4.3}$$

gives the one-step probability of transitions between all pairs of states. A row represents all possible fates given a particular starting point. The probabilities in each row add up to 1. A column represents all possible ways of arriving at any one state. Probabilities of each type of transition occurring in more than one step can be calculated using matrix multiplication. Formulating and iterating Markov chains for biological systems is discussed in detail in Jeffers (1978), Usher (1992) and Guttorp (1995).

A Markov set-chain is defined in terms of *parameter intervals*. The boundaries of the intervals are contained in two matrices,

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix} \text{ and } \mathbf{Q} = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1n} \\ q_{21} & q_{22} & \cdots & q_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ q_{n1} & q_{n2} & \cdots & q_{nn} \end{bmatrix}, \qquad (4.4)$$

where p_{ij} is the smallest chance and q_{ij} is the largest chance of making the transition from state i to state j in one step. The rows of the matrices **P** and **Q** do not sum to 1. In any given year the actual transition probability may be any value within the interval, including the minimum or maximum. However, the combinations of parameters that can co-occur are limited. The *actual* matrix A of transition probabilities for any given year is a regular Markov chain. Each parameter, a_{ij}, in the matrix has a value within the corresponding interval, (p_{ij}, q_{ij}), and the elements of each row do add up to 1. This limits the combinations of parameter values so that, for example, every parameter cannot achieve its maximum value in the same year.

There are various approaches for iterating and finding asymptotic behavior of Markov set-chains. Transition probability intervals for more than one step may be calculated directly by finding the geometric vertices of the distribution-space and iterating over successive time steps (Hartfiel 1998, Samuels and Kot 2001, in review). Unfortunately, this exact method is computationally unwieldy (Hartfiel 1998). Alternatively, and more practically, distribution intervals may be estimated using the Hi-Lo method (Hartfiel 1991, 1998, Samuels and Kot 2001, in review) or by Monte Carlo simulation (Samuels and Kot 2001, in review). In either case, the resulting predicted

distribution interval gives minimum and maximum boundaries on the frequency of each state within the community at each time. The asymptotic distribution interval is independent of initial conditions. The frequency of individuals in each state at time *t* is some value on the *distribution interval* between its minimum and maximum. The actual community composition in any year is described by a vector made up of the frequency of individuals in each state. The sum of the frequencies for all states is 1.

The widths of the intervals are important. There is no restriction on interval width. The narrowest possible parameter interval is a single value where the upper and lower boundaries are at the same point and the transition probability is precisely that value. The widest possible interval is where a transition probability ranges from 0 to 1. Clearly, however, a very wide interval is information-poor, and thus cannot be expected to deliver precise predictions. In the worst-case scenario a model, painstakingly parameterized, delivers the less-than-enlightening prediction that each state will make up between 0 and 100% of the system. Luckily, as we will see, this situation is the exception rather than the rule.

Set-chain model of grassland restoration

Throughout this discussion I draw upon an example from grassland restoration, using data from a prairie restoration experiment conducted at The Land Institute (TLI) in central Kansas. Native perennials were seeded on former agricultural land in four replicate plots. Species composition was recorded on all plots annually in July 1994-1999. Piper and Pimm (2000, in press) give a detailed account of the experiment. At its simplest, restoration of former agricultural land to prairie can be viewed as a process of

native perennials replacing weedy annuals (Holt et al. 1995). Perennial grasses establish themselves first, followed by forbs. This pattern of directional replacement of annuals by perennials is evident in the time series from the experiment. In native prairie, annuals represent less than 1% of cover (Piper, 1995). In the plots at TLI, initially up to 70% of plant cover was annual. By 1999 all plots showed an increase in perennials to 85-95% of total cover.

The aim of this paper is to use simple Markov set-chains to examine attributes of set-chains as models for ecological systems. I begin with the most basic set-chain and then add complexity, considering a two-state (perennials, annuals), three-state (perennial graminoids, other perennials, annuals), and five-state (C3 graminoids, C4 grasses, composites, legumes, other perennials, annuals) model (Figure 4.1). The parameters of the set-chains used throughout this paper are taken from data from the plots planted with sixteen species in the experiment.

Regular Markov chain parameters are individual-level replacement probabilities. The experimental data set comprises time series describing the frequency $n_{j,t}$ of each functional group j in the community at each time step t. To construct a Markov chain we need to know the probability that an individual in any state will be replaced by an individual of each other state (or stay the same) over one time step. However, most ecological data sets simply do not include this information. We know the frequency of individuals in each state each year. However, the time series does not specify what replacements occur in which combinations to generate the changes we see. Data based on frequencies but not individual transitions, termed aggregate or macro data (Lee, Judge

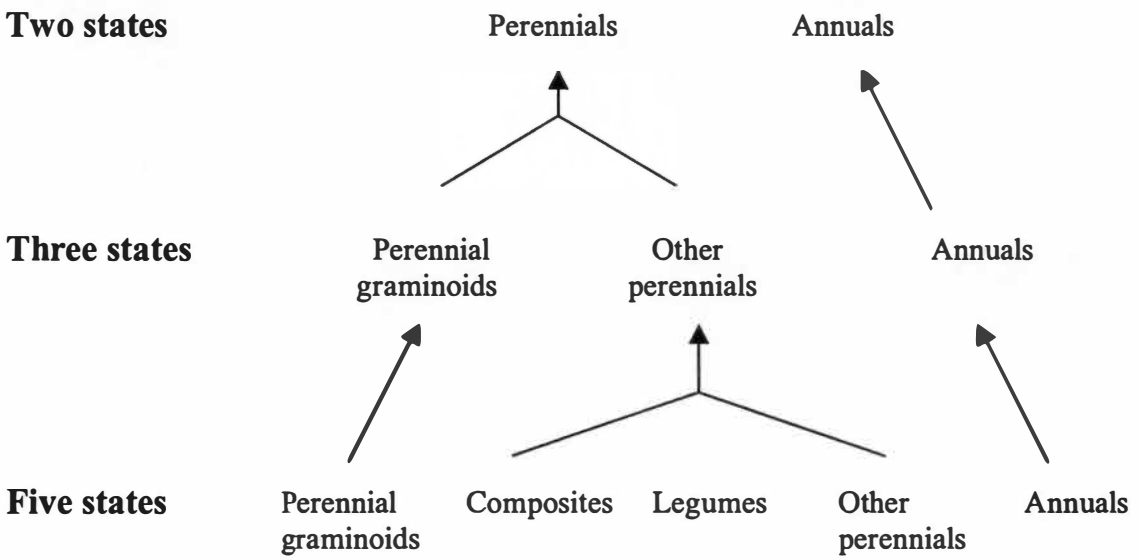


Figure 4.1: Three different discrete state representations of a grassland system. Two functional groups are expanded into three and five.

and Zellner 1977), is fairly common when dealing with ecological succession, making deriving Markov chain parameters a challenge (Usher 1992). We can, however, use the information in the time series to estimate the transition probabilities of the Markov chain that best describes the system dynamics.

To estimate transition probabilities from the aggregate grassland data, I used an unweighted least squares method. It is a search method that finds the Markov chain that best accounts for the given time series. Miller (1952) first proposed using least squares methods to estimate transition probabilities from aggregate data. The approach was further developed by and Lee, Judge and Zellner (1978) for use in economics and by Cooke (1981) for constructing Markov chain models of ecological succession. The estimated Markov chain does describe the system, but we should keep in mind that the estimation is a potential source of error for the set-chain calculations.

The data set provides a time series $y_{j,0}$, $y_{j,1}$, ...,$y_{j,t}$, , ...,$y_{j,T}$, the predicted frequency of individuals in state j over a series of successive time steps from 0 to T. Given a time series, we can find the transition probabilities a_{ij} that best describe it. The actual observed frequency $y_{j,t}$ and the predicted frequency $b_{j,t}$ (4.2) differ by some error,

$$u_{j,t} = y_{j,t} - \sum_i y_{i,t-1} a_{ij} \,. \tag{4.5}$$

Y is the matrix of observed frequencies,

$$\mathrm{Y} = \begin{bmatrix} y_{1,0} & y_{2,0} & \cdots & y_{n,0} \\ \vdots & \vdots & \ddots & \vdots \\ y_{1,t-1} & y_{2,t-1} & \cdots & y_{n,t-1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{1,T-1} & y_{2,T-1} & \cdots & y_{n,T-1} \end{bmatrix}, \tag{4.6}$$

and T is the time series length. T must be greater than the number of states, n, for there to be a unique solution.

The error between the observed sample proportions and the estimate is

$$\mathbf{u}_j = \mathbf{y}_j - \mathbf{Y}\mathbf{a}_j, \tag{4.7}$$

where $\mathbf{y}_i$ is the vector of sample proportions, $\mathbf{a}_j$ is a vector of transition probabilities to be estimated, and $\mathbf{u}_j$ is the vector containing error or disturbance on the frequency of each state. We wish to find $\mathbf{a}_j$ such that $\mathbf{u}_j$ is minimized. The difference between observed and calculated values in (4.7) can be expressed in quadratic form as the scalar sum of squares

$$\Phi = \mathbf{u}\mathbf{u}^T = (\mathbf{y}\ \mathbf{Y}\mathbf{a})(\mathbf{y}\ \mathbf{Y}\mathbf{a})^T, \tag{4.8}$$

where T denotes the transpose.

This unweighted least squares method was used to estimate a Markov chain for each of the four replicate plots. Fortran code for a quadratic programming method of least squares estimation of Markov chains was obtained from Lee (2000). Set-chains were constructed by taking the minimum and maximum observed value of each transition probability in the four replicate plots (Figure 4.2). Each set-chain is an estimated model based on data from four replicate plots observed over six years.

Uncertainty of parameters and predictions

Any data set captures only a part of all possible community dynamics. The forces driving community change vary from site to site and year to year. Markov set-chains incorporate this variation in their design, allowing us to model a fluctuating system. However, accommodating uncertainty comes at a cost. Imprecision in the parameter

Minimum **Maximum**

(a)

$$\begin{array}{c} \\ \text{P} \\ \text{A} \end{array}\begin{pmatrix} \text{P} & \text{A} \\ 0.941 & 0.009 \\ 0.433 & 0.442 \end{pmatrix} \qquad \begin{array}{c} \\ \text{P} \\ \text{A} \end{array}\begin{pmatrix} \text{P} & \text{A} \\ 0.991 & 0.059 \\ 0.558 & 0.567 \end{pmatrix}$$

(b)

$$\begin{array}{c} \\ \text{G} \\ \text{O} \\ \text{A} \end{array}\begin{pmatrix} \text{G} & \text{O} & \text{A} \\ 0 & 0.557 & 0 \\ 0.242 & 0.440 & 0.012 \\ 0.080 & 0.298 & 0.436 \end{pmatrix} \qquad \begin{array}{c} \\ \text{G} \\ \text{O} \\ \text{A} \end{array}\begin{pmatrix} \text{G} & \text{O} & \text{A} \\ 0.443 & 1 & 0.081 \\ 0.505 & 0.727 & 0.099 \\ 0.234 & 0.416 & 0.565 \end{pmatrix}$$

(c)

$$\begin{array}{c} \\ \text{G} \\ \text{C} \\ \text{L} \\ \text{O} \\ \text{A} \end{array}\begin{pmatrix} \text{G} & \text{C} & \text{L} & \text{O} & \text{A} \\ 0.038 & 0.470 & 0 & 0 & 0 \\ 0.170 & 0.403 & 0.017 & 0.010 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0.067 & 0.060 & 0.109 & 0.232 \end{pmatrix} \qquad \begin{array}{c} \\ \text{G} \\ \text{C} \\ \text{L} \\ \text{O} \\ \text{A} \end{array}\begin{pmatrix} \text{G} & \text{C} & \text{L} & \text{O} & \text{A} \\ 0.397 & 0.962 & 0.245 & 0.040 & 0.054 \\ 0.491 & 0.696 & 0.276 & 0.065 & 0.089 \\ 1 & 1 & 0 & 0.035 & 0.363 \\ 1 & 0.563 & 0 & 0 & 1 \\ 0.171 & 0.229 & 0.253 & 0.286 & 0.565 \end{pmatrix}$$

Figure 4.2: Markov set-chain models of grassland restoration with (a) two states, (b) three states, and (c) five states. Intervals are increasingly wide for systems with more states, because more different Markov chains can describe the dynamics of the systems with more types of transitions. Functional group codes are P = perennials, A = annuals, G = perennial graminoids, O = other perennials, C = composites, L = legumes.

intervals propagates into uncertainty about the distribution of individuals among states at future times.

While we expect uncertainty to propagate from parameters to predictions, exactly how the width of parameter intervals influences the width of the predicted distribution interval is not a simple matter. For example, Figure 4.3 shows three different two-state set-chains taken from the grassland experiment. With two states, each set-chain has four parameter intervals. The parameter intervals of any row are of equal width, but the rows are independent of one another. I used the Hi-Lo method (Appendix) to find the asymptotic distribution of each set-chain. We can compare predicted distribution intervals among these three set-chains. In all cases, the predicted distribution interval is wider than the average parameter interval. However, in (4.3a) and (4.3b) the widest parameter interval is actually broader than the distribution interval. This means that it is possible to know at least one parameter with less precision than is desired in the prediction. Indeed, in this situation the benefits of knowing *some* parameter precisely are great. While (4.3b) and (4.3c) have similar average parameter intervals, for (4.3c) where the uncertainty about parameters is distributed more evenly the predicted distribution interval is much wider.

In some cases narrowly delineated parameters can generate quite imprecise predictions. The two-state set-chain in (4.3c) predicts that somewhere from 0 to 53% of the community will be made up of annuals. It is doubtful that this broad a prediction is of any practical use for ecosystem management (or tells anything we couldn't have predicted without the benefit of a model!) Certainly, the predicted intervals can be very

	Set-chain				Parameter intervals		Predicted asymptotic distribution				
	Min		Max				Min		Max		Interval
	A	P	A	P	By row	Average	A	P	A	P	
(a)	0.442	0.433	0.567	0.558	0.125	0.051	0.015	0.880	0.120	0.985	**0.105**
	0.009	0.941	0.059	0.991	0.088						
(b)	0.439	0.330	0.670	0.561	0.231	0.150	0.000	0.828	0.172	1.000	**0.172**
	0.000	0.931	0.069	1.000	0.069						
(c)	0.717	0.126	0.874	0.283	0.157	0.151	0.000	0.465	0.535	1.000	**0.535**
	0.000	0.855	0.145	1.000	0.145						

Figure 4.3: Three Markov set-chain models with different parameter interval widths. Predicted distribution intervals are different for each set-chain. In both (a) and (b) one parameter interval is wider than the predicted distribution interval. While (b) and (c) have similar average parameter intervals, the predicted distribution interval for (b) is much wider. This suggests that the width of each parameter interval individually impacts the distribution interval size. A = annuals and P = perennials.

broad. How frequently does this occur? What attributes of a set-chain make for the broadest or tightest predictions? Does it matter whether all parameter intervals are specified with equal precision? What happens when we can measure some parameters to fairly precise values, but define others only by broad intervals? I used simulations to investigate the relationship between the width of the parameter intervals and the width of the predicted distribution interval.

For a preliminary view of how prevalent very wide predicted distribution intervals are, I examined interval widths in randomly constructed Markov set-chains. Random set-chains were constructed by selecting parameters from a uniform distribution between 0 and 1 for **P** (low) and **Q** (high) matrices. Parameters were restricted to combinations of values such that (a) $\mathbf{P}_{ij} < \mathbf{Q}_{ij}$ and (b) intervals were *tight* (Hartfiel 1998, Samuels and Kot Samuels and Kot 2001, in review). Boundaries on the asymptotic distribution were calculated for each set-chain using the Hi-Lo method (Hartfiel 1998, Samuels and Kot 2001). This was repeated for 10,000 set chains of each size two, three and four states. Interval width was recorded for one parameter and one state distribution for each set-chain. The interval to be measured was selected by generating a random integer between 1 and the number of parameters (or between 1 and the number of states, for the distribution interval).

The frequencies of different interval widths for the three-state set-chains are shown in Figure 4.4. The distribution of interval widths for the two- and four-state systems was nearly identical to those for the three-state systems. Over 60% of all parameter intervals in the randomly generated set-chains are narrower than 0.3. The

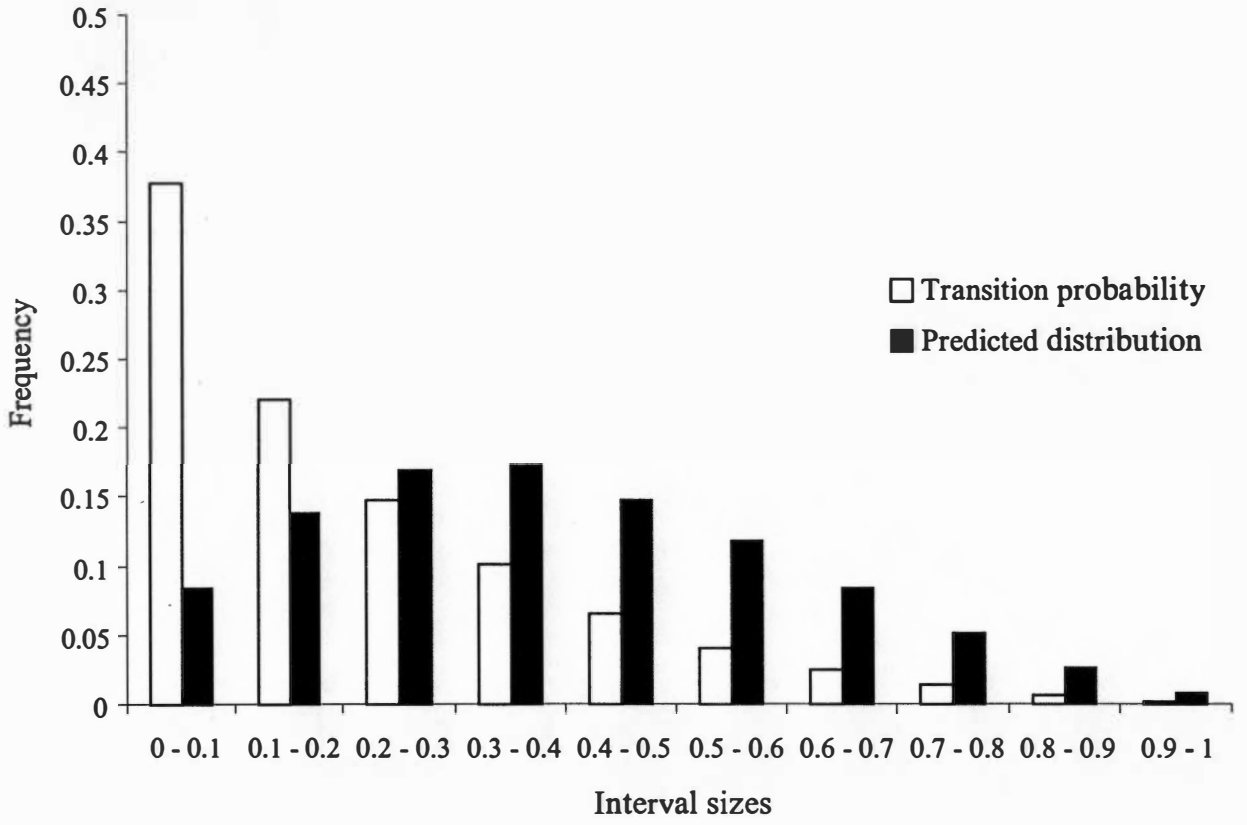


Figure 4.4: Parameter intervals and predicted distribution intervals in 10,000 randomly generated three-state Markov set-chains. Narrow parameter intervals are fairly common, with over 55% of all intervals narrower than 0.2. Predicted distribution intervals wider than 0.7 are rare, and those narrower than 0.4 are most common. Extremely wide prediction intervals are not necessarily inherent to the set-chain approach.

wider the interval, the rarer it is. This is a consequence of distributing probability-space over several states; if a parameter interval for one element of a row is very wide, then the intervals for the remaining elements are forced to be narrow. While predicted distribution intervals are usually wider than the parameter intervals, over 50% have widths between 0.2 and 0.5. Fewer than 5% have distribution intervals wider than 0.8. This suggests that extremely broad prediction intervals are not so common as to make Markov set-chains generally impractical for use.

Examining these random set-chains gives an idea of the average frequency of different interval widths. However, the relationship between specific parameter intervals and predicted distribution intervals is more complex. Even a small, three-state set-chain has nine parameters and millions of potential combinations of interval widths. To get a better idea of how the parameter interval widths affect the distribution intervals, I constructed random set-chains with controlled parameter interval widths. Table 4.1 shows the frequencies of predicted interval widths for a three-state, nine-parameter, system, where different numbers of parameter intervals (from all nine to zero) are restricted to widths narrower than 0.1. Additionally, I considered random set-chains where all nine parameters had intervals restricted to widths less than .2 and .3. Table 4.1 shows that as the number of more precisely known parameters decreases, the width of the predicted distribution increases. Interestingly, though, it appears that it is better to know all parameters within a .2 or .3 range than to know two thirds of them within a .1 range.

What does this say about Markov set-chains as models for real ecological systems? In natural systems a variety of different situations may lead to imprecision in

Table 4.1: Distribution of predicted interval sizes for Markov set-chains with different numbers of precisely known parameter intervals. The number of restricted parameters is the number of nine possible parameters that were limited to sizes equal or less than the specified parameter interval size.

Parameter interval size	# of restricted parameters	Predicted interval size									
		0-.1	.1-.2	.2-.3	.3-.4	.4-.5	.5-.6	.6-.7	.7-.8	.8-.9	.9-1
<.1	9	0.53	0.22	0.10	0.06	0.03	0.02	0.02	0.01	0.01	0.00
<.1	8	0.50	0.24	0.12	0.07	0.03	0.02	0.01	0.01	0.00	0.00
<.1	7	0.36	0.27	0.14	0.09	0.05	0.03	0.02	0.02	0.01	0.00
<.1	6	0.27	0.25	0.15	0.12	0.08	0.06	0.04	0.02	0.01	0.00
<.1	5	0.22	0.25	0.18	0.12	0.10	0.06	0.04	0.02	0.01	0.00
<.1	4	0.20	0.22	0.18	0.13	0.11	0.08	0.04	0.03	0.02	0.00
<.1	3	0.14	0.20	0.19	0.16	0.12	0.08	0.05	0.04	0.02	0.01
<.1	2	0.12	0.18	0.19	0.15	0.13	0.09	0.07	0.05	0.02	0.01
<.1	1	0.08	0.15	0.20	0.17	0.14	0.10	0.08	0.04	0.02	0.00
<.1	0	0.08	0.14	0.16	0.18	0.16	0.11	0.09	0.05	0.02	0.01
<.2	9	0.32	0.35	0.16	0.08	0.05	0.02	0.02	0.01	0.00	0.00
<.3	9	0.21	0.30	0.23	0.13	0.07	0.03	0.02	0.01	0.00	0.00

parameters. In some cases, like our grassland data, we might have fairly precise parameter estimates, but want to account for some degree of fluctuation or error that is fairly uniform across parameters. On the other hand, a scenario might arise where we have good measurements of some parameters, but little information about others. This could occur for a system where actual transitions were observed, but some state or states was not measured. Set-chains can operate in either of these situations, but it is important to keep in mind that one or a few imprecisely known parameters can have a big impact on the width of the predicted distribution interval. Of course, it is also good to keep in mind that precise parameter estimates are only as valuable as they are accurate. A set-chain's predictions depend on having accurately measured parameter intervals

Calculating boundaries vs. simulating dynamics of set-chains

As an alternative to Hartfiel's (1991, 1998) Hi-Lo algorithm, Monte Carlo simulations may be used to estimate the boundaries on the distribution of individuals among states from a Markov set-chain. (Samuels and Kot 2001, in review). There are trade-offs between the Hi-Lo method and the simulation methods. Both are computationally fairly simple. The Hi-Lo method gives an estimate of the boundaries on the predicted distribution but does not tell anything about the frequency of different values within them. The simulations provide a view of the range of different possible community compositions over a large number of runs. Additionally, with simulations it is possible to explore the consequences of different types of distributions of parameters within the intervals. However, simulation does not guarantee a perfectly accurate estimate of the boundaries, whereas the Hi-Lo method does.

To compare assumptions and predictions between the Hi-Lo method and simulation methods, I consider predicted asymptotic distributions for the set-chains in Figure 2.4 from three different estimation methods. I used the Hi-Lo method and Monte Carlo simulations with two different types of distributions, a uniform distribution and a bootstrapped distribution. I first describe how each method proceeded, and then compare and discuss differences between the methods and their predictions.

Hi-Lo method

The Hi-Lo method (Hartfiel 1991, 1998) is explained, with a detailed example, in Samuels and Kot (2001, in review). Java code for the simulations in this chapter is given in the Appendix. The Hi-Lo method is a modified matrix multiplication approach, a method for constructing rows and columns to multiply together that give the minimum and maximum value for each element of the next-step matrix. The Hi-Lo predictions are expressed as two matrices, one containing all the lowest possible probabilities of making the transition in the given number of time steps, and the other containing all the highest possible transition probabilities. A predicted distribution interval is then found by multiplying the multi-step transition matrices by the minimum and maximum boundaries on a specified initial conditions interval. Each boundary on each parameter interval converges to a stationary value in the limit of large time.

Monte Carlo simulation with uniformly distributed parameter values

A Markov set-chain may be simulated by iterating an inhomogeneous Markov chain with each transition probability chosen from a uniform distribution within its respective parameter interval at each time step. There is one restriction on the

construction of these random matrices. Because the rows must sum to 1, after choosing all the parameters are chosen, one element of each row must be adjusted so that the row sums to 1. Each step, one parameter is selected to be the one that is adjusted. Selection is by generating a random integer between 1 and the number of states. The distribution among states at each time is then calculated by matrix multiplication. This process generates a time series of distributions among states at each time step. The distribution fluctuates over time. Over the course of the simulation, I recorded the minimum and maximum frequency of each state. The highest and lowest observed frequencies give an estimate of the boundaries on the possible distributions of individuals among states in the system.

Monte Carlo simulation with bootstrapped distribution of parameter values

The second approach I use to simulate Markov set-chains is a modified bootstrapping method. Bootstrapping (Efron 1979, Manly 1997) is a method of resampling with replacement. It can be used to estimate the sampling distribution of parameter estimates from limited data. We have only four measurements of each transition probability. However, the best representation we have of the distribution of transition probabilities within the interval boundaries in the system is whatever limited data we have. So, instead of drawing probabilities from a uniform (or other) distribution, we simulate by drawing probabilities directly from our set of estimated transition probabilities. Note that while the simulation here was based on a bootstrapping approach, I have controlled the parameter values in the simulation to generate only set-chains with tight intervals. As in the uniform distribution simulation above, to ensure

that rows sum to 1, I adjusted one parameter in each row. To iterate, a matrix was constructed for each time step by drawing each parameter at random from the set of values for that parameter that were estimated from the data. The system was iterated by matrix multiplication, using the same method as in the uniform distribution simulation above, to generate a time series and record minimum and maximum frequencies of each state observed during the entire run.

Comparing methods

I wanted to see how accurately the simulation methods estimate the boundaries calculated by the Hi-Lo method. I used the Hi-Lo method and each simulation approach to find the asymptotic distribution interval boundaries on the set-chains in Figure 4.2. Results from the three different methods are compared in Table 4.2. Each method of finding boundaries on the asymptotic distribution of a set-chain has advantages and disadvantages. Both the Hi-Lo method and the uniform distribution Monte Carlo simulations calculate boundaries based on information about the minimum and maximum matrices, but their predictions provide different types of information. The Hi-Lo method gives absolute best- and worst-case predictions with no information about the distribution within them. The simulation does not find the absolute interval boundaries, but it does give information about the distribution of different community compositions. The fact that the uniform distribution simulations do not pick up the extremes indicates that, if parameter values are indeed uniformly distributed within their intervals, these extremes occur only very rarely. In 10,000 runs, where all transition probabilities within an

Table 4.2: Boundaries on the predicted distribution of individuals among states after 100 steps from 10,000 runs for (a) two-state, (b) three-state, and (c) five-state systems using three methods. The estimate from bootstrapped parameter simulations is closer to the Hi-Lo method predictions than the estimate based on uniformly distributed parameters. However, both simulation methods differ greatly from the Hi-Lo predictions for a large system with broad interval predictions (five-state annuals).

		Hi-Lo method		Monte Carlo simulation with uniformly distributed parameter values		Monte Carlo simulation with bootstrapped parameter values	
		Minimum	**Maximum**	**Minimum**	**Maximum**	**Minimum**	**Maximum**
(a)	**Annual**	0.0152	0.1203	0.0252	0.1036	0.0173	0.1138
	Perennial	0.8797	0.9848	0.8964	0.9748	0.8862	0.9827
(b)	**Annual**	0.0114	0.1793	0.0385	0.1418	0.0178	0.1494
	Perennial graminoid	0.1131	0.4933	0.1513	0.4456	0.1257	0.4807
	Other perennial	0.4287	0.8569	0.4675	0.7723	0.4472	0.8293
(c)	**Annual**	0.0000	0.5019	0.0371	0.1881	0.0036	0.1879
	Perennial graminoid	0.0260	0.6586	0.1212	0.4414	0.0971	0.5370
	Composite	0.1229	0.8926	0.3379	0.6663	0.3160	0.7834
	Legume	0.0047	0.2714	0.0216	0.2175	0.0092	0.2468
	Other perennial	0.0027	0.1697	0.0177	0.0795	0.0074	0.0848

interval were equally likely, the combinations of parameter values that lead to the most extremes behavior never came up.

The bootstrapped distribution simulation, on the other hand, uses different information. It takes into account all the specific estimated transition probabilities rather than only the boundaries, leading to predictions closer to those of the set-chain. However, this is not surprising because the mechanics of the bootstrap are also closer to those of the set-chain. With the bootstrapping approach, the extreme values of each transition probability are nearly guaranteed to occur (since with four replicates each occurs with probability 25%). If the observed values are a good indication of the typical behavior of the system, then the extremes of the Hi-Lo interval predictions are not as rare an occurrence. Since the bootstrap predictions match those of the Hi-Lo method fairly well, bootstrapping may be a useful and computationally simpler alternative to the Hi-Lo method for finding absolute boundaries. However, particularly with only four replicates, the realized distribution of transition probabilities from year to year is probably somewhere between a uniform distribution and a bootstrapped distribution.

Even though the Hi-Lo method is most accurate at finding boundaries, the other simulation methods have other advantages. The Monte Carlo methods give information about the distribution within the interval we do not get using the Hi-Lo method. These help give a view of what behavior is most likely to occur. With further development, it may be possible to integrate information from both approaches to predict boundaries with confidence intervals or most likely system configurations.

Fluctuating parameters in a finite world

Markov set-chains are designed to deal with varying parameter values. Still, the accuracy of set-chain predictions relies on having correctly specified interval boundaries. How much data is required to provide a good representation of the system? The grassland data set is limited in terms of time series length, number of replicates and the level of detail of the observations (number of states). These factors dictate how much we know about the system and how confident we can be about parameter interval estimates. To understand how faithfully we can expect the set-chain estimated from a data set to represent the system dynamics, I experimented with estimating parameter intervals using simulated data from predetermined, *given* set-chains.

Simulating data from set-chains

To experiment with estimating parameters, I used a given set-chain for each of three system sizes. I simulated data using, as the given set-chains, the grassland set-chains from plots planted with sixteen species (Figure 4.2). I used the uniformly distributed Monte Carlo simulation method described above to generate ten "replicate" twenty year time series for the two-, three- and five-state set-chains. Shorter time series, of four, eight, twelve and sixteen years, were then obtained by taking segments from the beginning of each twenty-year series *(i.e.* each four-year time series is the first four years of a twenty-year series). The method for estimating Markov chains from time series data (Lee, Judge and Zellner, 1977) requires at least one more year of data than there are states. Thus, for the five-state system the shortest time series was six years instead of

four. Twenty years was the longest series I generated, because it was the longest five-state series the estimation program (Lee, 2000) would allow.

My goal was to assess how the accuracy of parameter estimates for Markov set-chains varies with system size, time series length, and number of replicates. To do this, I estimated Markov set-chains from different subsets of the simulated data and measured how close each estimate was to the set-chain that generated it. The simulated data set comprises ten replicate time series of each length for each system size. I used an estimation program (Lee, 2000) to find the regular Markov chain that best describes each time series. This resulted in, for each time series length and system size, a set of ten different Markov chains, each of which can contribute to an estimate of the set-chain that generated the data.

Measuring distance between given and estimated set-chains

For each combination of time series length and system size, I considered how adding replicates affects the estimated set-chain. To do this, I estimated set-chains based on Markov chains from up to ten replicates. I began with only one Markov chain (a set-chain where minimum and maximum parameters are equal and interval widths are zero), and then incorporated information from one additional Markov chain at a time. As each was added, if a value in the newly added Markov chain was either less than the minimum or greater than the maximum previously specified, I updated this parameter in the new set-chain. I then calculated the distance of each estimated Markov set-chain from the actual set-chain, to compare the accuracy of the estimate for different data sets.

I calculated the distance between the estimated and given set-chains as the Euclidean norm of the difference between all elements of the set-chain; both the minimum and the maximum boundaries were used in computing distance. However, in reporting these distances (Figure 4.5, 4.6) I distinguish between two components of distance, underestimating and overshooting of boundaries. The estimate for any parameter may fall on either side of its value in the given set-chain, and the effect on the estimated interval is different. Underestimating the boundaries (estimating a minimum value that is greater than actual minimum or estimating a maximum less than the given maximum) results in an estimated interval that is narrower than the interval in the given set-chain. Predictions from a set-chain with underestimated parameter intervals could fail to include some possible system dynamics. On the other hand, parameter estimates that overshoot the boundaries will lead to wider intervals than in the given set-chain. While wider intervals may be undesirable as they could generate less precise predictions, the predicted community composition does *at least* capture the range of dynamics that is possible from the original given set-chain.

Distances between the estimated and given set-chains are shown in Figure 4.5 (underestimate) and 4.6 (overshooting). In both cases, the distances are expressed as the average distance per parameter, in order to compare distance across all systems sizes. Table 4.3 is a summary of the distances between the given set-chains and the best estimates from ten replicates. This shows the effects of system size and time series length. In table 4.3 I have also included the maximum distance of any one parameter from its given value. Differentiating between average distance per parameter and the

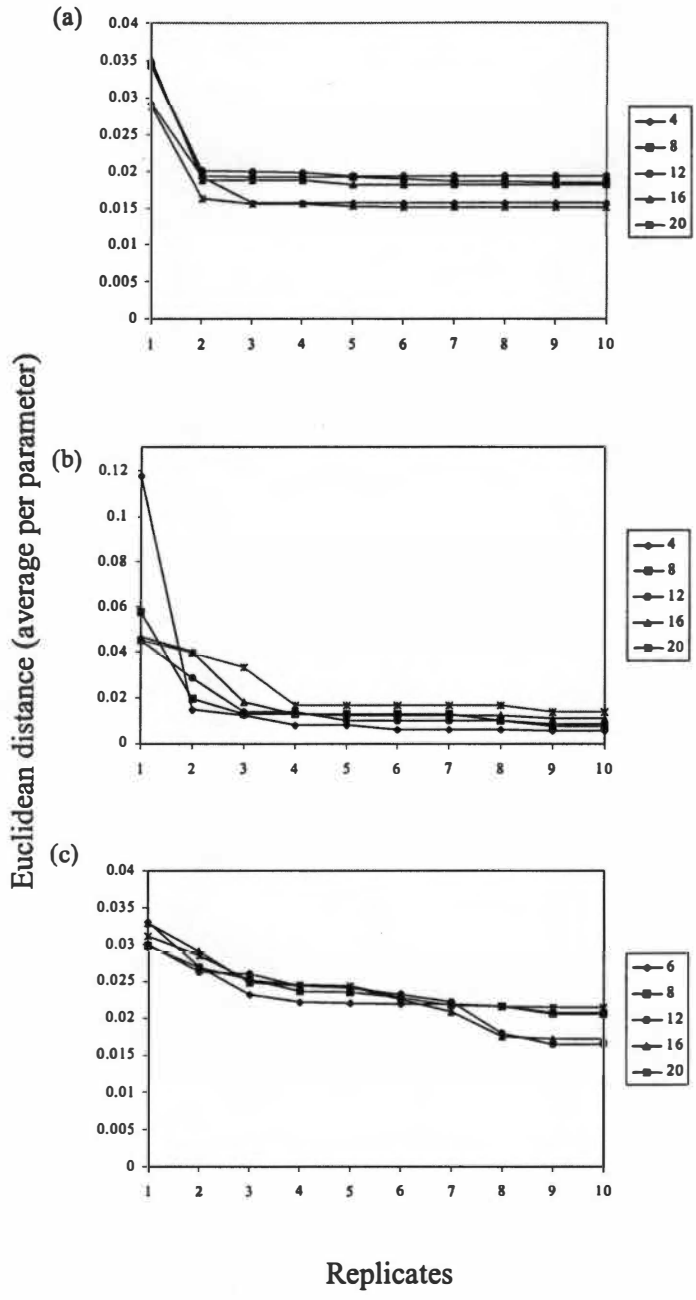


Figure 4.5: Euclidean distance of estimated Markov set-chains from true set-chain parameters for a system with (a) two states, (b) three states, (c) four states. Note that the scale for the three-state set-chain is different from the others. Adding replicates beyond four does not improve the estimate. For all system sizes, the average distance with ten replicates is near 0.02.

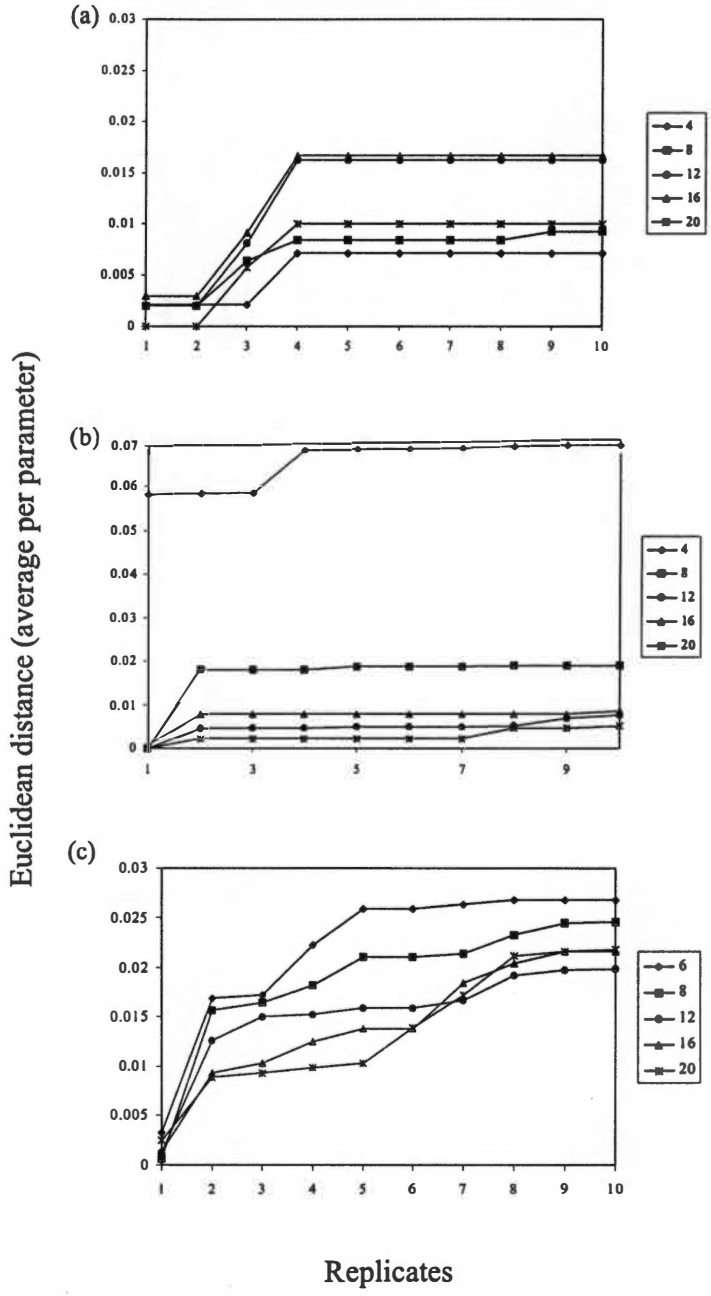


Figure 4.6: Euclidean distance of estimated Markov set-chains in excess of true set-chain parameters boundaries for a system with (a) two states, (b) three states, (c) four states. Note that the scale for the three-state set-chain is different from the others.

greatest distance of any one parameter helps show how much of the entire distance can be due to error in estimating one parameter.

Evaluating data requirements

Overall, the distances of estimated parameters from the given parameter values are small relative to the magnitude of the parameters; average underestimation distances (Figure 4.5) are on the order of .02, whereas the parameters being estimated have values up to .99. Overshooting of interval boundaries appears to be a greater source of estimation error than underestimation. As we add replicates we quickly achieve the best estimate we can expect with regard to underestimating interval boundaries and width (Figure 4.5). However, as we add more data, the magnitude of overestimation, estimating intervals that are wider than the original given intervals, increases (Figure 4.6).

Of the factors varied in the simulations, system size has the greatest impact on the accuracy of estimates of Markov set-chains. The estimation method simply does not work well for large systems. In the five-state system estimated parameters are often as far from their correct value as possible; a zero is estimated as a 1 (Table 4.3). However, this problem may be exacerbated by the specific parameters of the five-species system we are estimating; it has broader intervals and more zeroes than either the two- or three-state system. Additionally, it may be more difficult to estimate parameters for a more complex description of a system because not all groups interact with one another and there are more possible ways to get the same outcome. Even though the search method finds the best Markov chain, there are many different Markov chains that come close to the best estimate. A further difference between the system sizes arises in the method I used to

Table 4.3: Distances of best estimate from ten replicates from true set-chain parameter interval boundaries by three measures for three system sizes. Codes for distances are: E = Average per parameter Euclidean distance, and M = Maximum distance of any one parameter in the set-chain. (a) Underestimation, distance inside the true boundary. (b) Overshooting, distance beyond the true boundary. (c) Absolute distance, regardless of which direction, from the true boundary. Underestimate and overshoot distances may cancel each other out in absolute distance. Overshooting boundaries contributes more to error than underestimating.

	Years	States					
		2		3		5	
		E	M	E	M	E	M
(a)		**Distance short of boundary**					
	4	0.016	0.048	0.006	0.002	0.021	1.000
	8	0.018	0.070	0.008	0.006	0.021	1.000
	12	0.019	0.072	0.008	0.006	0.016	0.757
	16	0.018	0.069	0.011	0.007	0.017	0.821
	20	0.015	0.054	0.014	0.007	0.021	1.000
(b)		**Distance beyond boundary**					
	4	0.007	0.048	0.069	0.557	0.027	0.783
	8	0.009	0.070	0.019	0.147	0.025	0.836
	12	0.016	0.072	0.008	0.054	0.020	0.757
	16	0.017	0.069	0.009	0.048	0.020	0.765
	20	0.010	0.054	0.005	0.032	0.022	0.807
(c)		**Absolute distance from boundary**					
	4	0.009	0.048	0.043	0.557	0.031	1.000
	8	0.010	0.070	0.004	0.147	0.029	1.000
	12	0.013	0.072	0.001	0.076	0.024	0.870
	16	0.012	0.069	0.043	0.501	0.025	0.906
	20	0.009	0.054	0.037	0.427	0.029	1.000

simulate the data. Because these time series are generated by Monte Carlo simulation, having more states means that more different possible time series could be generated. It takes six years of data to be able to estimate the Markov chain, yet 25 parameters are changing every year so it is less likely that a single Markov chain can do a good job of approximating what happened.

For three-state systems, the overall Euclidean distances (Table 4.3) are less than for two-state systems. This distance is accounted for by the smaller overshooting of boundaries in the estimates for two-state systems. The number of years of data has little effect on estimates for the two- or five-state systems (Figure 4.6). For three-state systems the distance of the estimate beyond the interval decreases with increasing time series length. I experimented with using Lee's (2000) estimation program to estimate regular Markov chains and found little improvement with increasing time series length. For both regular Markov chains and set-chains, a longer time series occasionally improves the estimate, but not reliably. Additionally, estimates of regular Markov chains are decreasingly accurate for increasing system size. This suggests that the low importance of time series length in my simulations may be a consequence of the estimation method I used. Even when a regular Markov chain generates the data, there is a limit to the accuracy of the estimate. With Markov set-chains a different Markov chain dictates system change during each step; there is not a straightforward relationship between time series length and information about transition probabilities.

Without having observed actual transitions (*micro* data) it may not be feasible to construct set-chains with large numbers of states. What is important for obtaining a good

estimate of parameter intervals appears to depend on the number of states. For the two-state system, it is fairly easy to get good estimates. In the three-state systems, using shorter time series gives a better guarantee of getting close to the boundaries, because more time leads to increased overshooting of the boundaries. Based on the simulations here the twelve-year time series seems the best for estimating the given matrix.

Summary and discussion

I have examined three sources of error in iterating and finding asymptotic distributions of Markov set-chains. The aim was to understand the impact of each on the overall utility of set-chains as models for succession. This work points to some guidelines as to what type of data is needed and what sorts of attributes of set-chains can lead to the most useful models.

In general, the issues that raised concerns about whether set-chains could be used to model ecological systems do not preclude their use as models for succession. First, I considered the impact of parameter interval widths. While knowing parameter values within the narrowest interval possible can help generate more precise predictions, it appears that the uncertainty in parameter intervals propagates less than might be expected. Even somewhat wide parameter intervals can generate useful predictions. Second, I looked at the uncertainty in methods of iterating set-chains. I considered the Hi-Lo method and two simulation methods. There were differences between the predictions of the three. Overall, if the interest is in finding the boundaries, the best estimate will come from the Hi-Lo method. However, simulations can provide a view of aspects of system dynamics other than boundaries on the distribution. Simulations using

more types of distributions might give added insight into the consequences of different types of parameter distributions and what types of distributions actually occur in nature.

I also considered how much and what sort of data is required to get a good estimate of the set-chain that generated it. This was important for telling whether Markov set-chain models may be used to study the dynamics of the grassland restoration experiment. I was concerned that the data might be too limited to give a good idea of what the parameter intervals are. Despite limitations on the time series length and number of replicates, this fairly typical data set does offer enough information to provide reasonable estimates of set-chain parameters.

Set-chains can be a useful approach for modeling succession in grasslands. The interval predictions give a new type of information that is not part of the predictions from regular Markov chains, the estimate of the breadth of the range of possibilities. This makes set-chains a useful tool for planning ecological restoration and construction. All the sources of error evaluated here remain important to keep in mind when using set-chains. It certainly helps to understand what their effects are. However, none of them is prohibitive. As is true for most modeling approaches, more information and better measurement improve the quality of the set-chain model. We have gained a clearer view of what factors influence the width and values of the intervals in Markov set-chains. This will help with constructing and interpreting resulting predictions from a set chain using data from real systems.

CHAPTER 5

CONCLUSION: MAKING UNCERTAINTY AN ASSET FOR ECOLOGICAL MODELING

Stochasticity in natural systems is often viewed as a complication and source of instability for ecological modeling (Chesson and Case 1986, DeAngelis and Waterhouse 1987). For modeling natural systems with Markov chains, fluctuating parameters are a particularly confounding problem (Usher 1981, Hobbs 1983). Markov set-chains turn this situation around, taking advantage of uncertainty and fluctuations as a source of information about variability. Set-chains model succession in a way that generates predictions about the widths of the ranges of possible species frequencies.

While contributing to ecological theory, set-chains also add a new tool to help guide applied restoration and management. In fact, Markov set-chains are an encouraging example of co-evolution of theoretical and empirical research. Discussion of the potential for theory to contribute to restoration practices has often focused on rules and patterns from community assembly studies (Hobbs and Norton 1994, Lockwood 1997, Lockwood and Samuels unpublished manuscript). The grassland experiment that generated the data for my set-chains was implemented as a test of predictions from models of community assembly. Now, the results from that experiment have been used in developing a new modeling approach.

The work presented in this volume is an introduction to set-chains and what they can offer ecologists. Set-chains are still relatively new, both to ecology (Samuels and Kot 2001, in review) and to the mathematical literature (Hartfiel 1998), so there are ample unexplored applications and implications. Further study of set-chains will be important to address questions about what they contribute. First, what do set-chains offer managers or restoration practitioners that can be put into practice immediately? Second, what theoretical areas can be expanded on and further investigated. Using set-chains now, and continuing to study them, will help understand what controls community variability, and will inform the practice of restoration.

Applying set-chains for restoration and management

Aspects of what we learn from set-chains can be incorporated immediately into the practice of ecological restoration and management. Set-chains give information about how introducing different species mixtures can affect the outcome of restoration. I have shown that changing transition probability intervals can drive a system from a situation where its outcome becomes less predictable over time to one where the range of possible outcomes is more certain. Introducing different species ensembles can change transition probabilities, and thus alter system dynamics. Finding out how particular species contribute to transition probabilities will require further investigation. However, it is important to be aware all functional group members are not equal. For now, restoration practitioners might experiment with including more species in their mixtures. Wherever possible, restoration projects should be designed to double as experiments to increase our understanding of how different species influence system transition probabilities.

Eventually, set-chains may generate specific enough predictions to guide us in managing for a particular desired combination of species. However, accurate set-chain models require accurate transition probability measurements. This sounds daunting, but understanding transition probabilities does not necessarily entail measuring or testing probabilities for all possible species and combinations. Field experiments can help understand what attributes of species or species mixtures influence transition probabilities. In the grassland data set, the effects on variability may be due to differences in diversity, or a product of the particular species involved. If diversity controls variability, we could control variability wholesale by increasing diversity. On the other hand, we might identify individual species that contribute particularly well to speeding succession. Either way, understanding what controls transition probabilities can lead to guidelines for what to introduce and in what combinations.

My analysis of a restricted Markov set-chain classifies variability dynamics into four types. However, simply knowing whether intervals widen or narrow does not tell the whole story. Widening and narrowing can occur in different degrees. The specific applied goals for the system in question determine how much the form of variability dynamics matters. Continually growing intervals are generally undesirable when the aim is to manage for a specific species configuration. For example, in Natural Systems Agriculture, we do not want to wait years to obtain the system we want, so initial interval widening may not be acceptable. On the other hand, in a situation such as functional restoration by soil stabilization, transient dynamics might be less important. Decisions

about specific needs and goals will require a cost-benefit approach; there may be cases where increased predictability is not worth the cost of adding another species.

Further development of set-chain theory

I have discussed methods for iterating set-chains by simulation, and used analytic methods to show how they can lead to different patterns of variability. Simulation and iteration let us observe behavior of set-chains for any particular system. Analytic approaches, on the other hand, help understand the variety of types of dynamics that set-chains generate. There are potential benefits to expanding on each set of methods.

Set-chains iterated by the Hi-Lo algorithm estimate the boundaries on species frequencies over time, but do not give information about the likelihood of different configurations within those boundaries. Simulation approaches may lead to ways of estimating confidence intervals about the predicted boundaries of set-chains. Set-chains may also be adapted to provide more information about the distribution of states within boundaries, or confidence intervals about the boundaries. This would involve combining information from Monte Carlo simulation and set-chains iterated by the Hi-Lo method. Then, set-chain predictions might include estimates of the most likely system states.

Analysis of restricted set-chains has given an idea of what types of dynamics can occur. The four different combinations of intervals widening and narrowing over time are patterns of changing predictability. Analyzing a restricted set-chain where both annuals and perennials have interval transition probabilities will help see whether there are still other possible types of dynamics. Additionally, exploring the dynamics of systems with more than two states could lead to other types of behavior.

Conclusion

Combining theoretical development and empirical testing of set-chain models can lead to quantitative guidelines for controlling variability in ecological systems. By managing transition probability intervals, it may be possible to control the widths of the ranges of species frequencies that can result from community restoration or construction. Knowing what sort of dynamics to expect depends on having accurately estimated transition probability intervals for the species or groups involved. Some of the most important future work will involve developing methods for estimating and cataloging effects of species and aggregate community traits on transition probabilities. Eventually, with reliable methods for estimating transition probabilities, we could create software or other automated approaches for restoration practitioners to use to generate predictions from more complex set-chains.

With what we know right now, we can incorporate into management practices a qualitative view of how variability thresholds may affect systems. As we experiment with this, our experiences may lead to prescriptive guidelines focusing on the effects of particular species or groups on transition probabilities. Considering additional data sets may give further insight into what controls transition probabilities. Restoration experiments done with patterns of variability in mind can contribute data and help enhance our understanding of what controls variability. New data and further theoretical advances can feed back to one another, making set-chains a significant contribution to improving restoration and management of plant communities.

REFERENCES

REFERENCES

ACEVEDO, M.F. (1981). On Horn's Markovian model of forest dynamics with particular reference to tropical forests. *Theor. Popul. Biol.* **19**, 230-250.

BINKLEY, C.S. (1980). Is succession in hardwood forests a stationary Markov process? *Forest Sci.* **26**, 566-570.

CASE, T.J. (1990). Invasion resistance arises in strongly interacting species-rich model competition communities. *Proc. Natl. Acad. Sci.* **87**, 9610-9614.

CASE, T.J. (1991). Invasion resistance, species build-up and community collapse in metapopulation models with interspecies competition. *Biol. J. Linn. Soc.* **42**, 239-266.

CHAPIN, F.S. III, WALKER, B.H., HOBBS, R.J., HOOPER, D.U., LAWTON, J.H., SALA, O.E. & TILMAN, D (1997). Biotic control over the functioning of ecosystems. *Science* **277**, 500-504.

CHESSON, P.L., & CASE, T.J. (1986). Overview: nonequilibrium community theories: chance, variability, history, and coexistence. In: *Community ecology.* (J. DIAMOND & CASE, T. J., eds.), pp. 229–239. New York, NY: Harper and Row.

CONNELL, J.H. & SOUSA, W.P. (1983). On the evidence needed to judge ecological stability or persistence. *Am. Nat.* **121**, 789-824.

COOKE, D. (1981). A Markov chain model of plant succession. In: *The Mathematical Theory of the Dynamics of Biological Populations II.* (HIORNS, R.W. & COOKE, D., eds.), pp. 230-247. New York, NY: Academic Press.

DEANGELIS, D.L., & WATERHOUSE, J.C. (1987). Equilibrium and nonequilibrium concepts in ecological models. *Ecol. Monog.* **57**: 1-21.

DOAK, D.F., BIGGER, D., HARDING, E.K., MARVIER, M.A., O'MALLEY, R.E. & THOMSON, D. (1998). The statistical inevitability of stability-diversity relationships in community ecology. *Am. Nat.* **151**, 264-276.

DOBSON, A.P., BRADSHAW, A.D. & BAKER, A.J.M. (1997). Hopes for the future: restoration ecology and conservation biology. *Science* **277**, 515-522.

DRAKE, J.A. (1990). Communities as assembled structures: Do rules govern pattern? *Trends Ecol. Evol.* **5**, 159-164.

DRAKE, J.A. (1991). Community assembly mechanics and the structure of an experimental species ensemble. *Am. Nat.* **131**, 1-26.

EFRON, B. (1979). Bootstrap methods: another look at the jackknife. *Ann. Stat.* **7**, 1-26.

GRIME, J.P. (1997). Biodiversity and ecosystem function: the debate deepens. *Science* **277**, 1260-1261.

GRIMM, V. & WISSEL, C. (1997). Babel, or the ecological stability discussions: An inventory and analysis of terminology and a guide for avoiding confusion. *Oecologia* **109**, 323-334.

GUTTORP, P. (1995). *Stochastic Modeling of Scientific Data.* London: Chapman and Hall.

HARTFIEL, D.J. (1981). On the limit set of stochastic products $x\mathbf{A}_1 \ldots \mathbf{A}_k$. *Proc. Amer. Math. Soc.* **81**, 201-206.

HARTFIEL, D.J. (1987). Computing limits of convex sets of distribution vectors $x\mathbf{A}_1 \ldots \mathbf{A}_r$. *J. Statist. Comput. Simul.* **38**, 1-15.

HARTFIEL, D.J. (1991). Component bounds on Markov set-chain limiting sets. *J. Statist. Comput. Simul.* **38**, 15-24.

HARTFIEL, D.J. (1998). *Markov Set-Chains.* Lecture Notes in Mathematics **1695**, Berlin: Springer-Verlag.

HOBBS, R.J. (1983). Markov models in the study of post-fire succession in heathland communities. *Vegetatio* **56**, 17-30.

HOBBS, R.J. & LEGG, C.J. (1983). Markov models and initial floristic composition in heathland vegetation dynamics. *Vegetatio* **56**, 31-43.

HOBBS, R.J. & NORTON, D.A. (1996). Towards a conceptual framework for restoration ecology. *Restor. Ecol.* **4**, 93-110

HOLT, R.D., ROBINSON, G.R. & GAINES, M.S. (1995). Vegetation dynamics in an experimentally fragmented landscape. *Ecology* **76**, 1610-1624.

HORN, H.S. (1975). Markovian properties of forest succession. In: *Ecology and Evolution of Communities* (CODY, M.L. & DIAMOND, J.M., eds.), pp. 196-211. Cambridge: Belknap Press of Harvard University Press.

HORN, H.S. (1976). Succession. In: *Theoretical Ecology: Principles and Applications* (MAY, R.M., ed.), pp. 187-204. Oxford: Blackwell Scientific.

HUBBELL, S.P. & FOSTER, R.B. (1987). The spatial context of regeneration in a neotropical forest. In: *Colonization, Succession and Stability* (GRAY, A.J., CRAWLEY, M.J. & EDWARDS, P.J. eds.), pp.395-412. Oxford: Blackwell Scientific.

HUGHES, J.B. & ROUGHGARDEN, J. (1998). Species Diversity and Biomass Stability. *Am. Nat.* **155,** 618-627

HUXEL G.R. & HASTINGS, A. (1999). Habitat loss, fragmentation, and restoration. *Restor. Ecol.* **7,** 309-315.

IVES, A.R., GROSS, K. & KLUG, J.L. (1999). Stability and variability in competitive communities. *Science* **286,** 542-544.

IVES, A.R., KLUG, J.L. & GROSS, K. (2000). Stability and species richness in competitive communities. *Ecol. Lett.* **3,** 399-411.

JONSSON, G.F. & TREFETHEN, L.N. (1998). A numerical analyst looks at the cutoff phenomenon" in card shuffling and other Markov chains, In: *Numerical Analysis 1997*, (GRIFFITHS, D.F., HIGHAM, D.J. & WATSON, G.A., eds.) pp. 150-178. New York, NY: Addison Wesley Longman.

JEFFERS, J.N.R. (1978). *An Introduction to Systems Analysis, With Ecological Applications.* Baltimore: University Park Press.

KANJI, G.K. (1993). *100 Statistical Tests.* London: Sage Publications

LAWTON, J.H. & BROWN, V.K. (1993). Redundancy in ecosystems. In: *Biodiversity and Ecosystem Function,* (SCHULZE, E.D. & MOONEY, H.A., eds.), pp. 255-270. Berlin: Springer-Verlag.

LEE, T.C., JUDGE, G.G. & ZELLNER, A. (1977). *Estimating the parameters of the Markov probability model from aggregate time series data.* New York: North-Holland Publishing Company.

LEE, T.C. (2000). Home page. http://vm.uconn.edu/~wwware/tclpersn.html.

LOCKWOOD, J.L. (1997). An alternative to succession: assembly rules offer guide to restoration efforts. *Restor. Manag. Notes.* **15**, 45-50.

LOCKWOOD, J.L. & SAMUELS, C.L. Unpublished manuscript. The role of process in ecological restoration: tackling the problems one step at a time.

LOGOFET, D.O., GOLUBYATNIKOV, L.L. & DENISENKO, E.A. (1997). Inhomogeneous Markov models for succession of plant communities: New perspectives on an old paradigm. *Biol. Bull.* **24**, 506-514.

LOREAU, M. (2001). Biodiversity and ecosystem functioning: recent theoretical advances. *Oikos* **91**, 3-17.

MANLY, B.F.J. (1997). *Randomization, Bootstrap and Monte Carlo Methods in Biology.* London: Chapman and Hall.

MAY, R.M. (1972). What is the chance that a large complex system will be stable? *Nature* **237**, 413-414.

MAY, R.M. (1973). *Stability and Complexity in Model Ecosystems*. Princeton, NJ: Princeton University Press.

MCAULIFFE, J.R. (1988). Markovian dynamics of simple and complex desert plant communities. *Am. Nat.* **131**, 459-490.

MCCUNE, B. & ALLEN, T. F. H. (1985). Will similar forest develop on similar sites? *Can. J. Bot.* **63**, 367-376.

MCGRADY-STEED, J., HARRIS, P.M. & MORIN, P. (1997). Biodiversity regulates ecosystem predictability. *Nature* **390**, 507-509.

MICHELI, F., COTTINGHAM, K.L., BASCOMPTE, J., BJORNSTAD, O.N., ECKERT, G.L., FISCHER, J.M., KEITT, T.H., KENDALL, B.E., KLUG, J.T. & RUSAK, J.A. (1999). The dual nature of community variability. *Oikos* **85,** 161-169.

MILLER, G.A. (1952). Finite Markov processes in psychology. *Psychometrika* **17,** 149-167.

NAEMM, S. (1998). Species redundancy and ecosystem reliability. *Conserv. Biol.* **12,** 39-45.

NAEMM, S. & LI, S. (1997). Biodiversity enhances ecosystem reliability. *Nature* **390,** 507- 509.

NEUBERT, M.G. & CASWELL, H. (1997). Alternatives to resilience for measuring the responses of ecological systems to perturbations. *Ecology* **78,** 653-665.

NIJS, I. & IMPENS, I. (2000). Biological diversity and probability of local extinction of ecosystems. *Func. Ecol.* **14,** 46-54.

O'ROURKE, J. (1998). *Computational Geometry in C.* Cambridge: Cambridge University Press.

PATTEN, B.C. (1975). Ecosystem linearization: an evolutionary design problem. *Am. Nat.* **109**, 529-539.

PICKETT, S.T.A. & PARKER, T.V. (1994). Avoiding the old pitfalls: opportunities in a new discipline. *Restor. Ecol.* **2**, 75-79.

PIMM, S.L. (1984). The complexity and stability of ecosystems. *Nature* **307**, 321-326.

PIMM, S.L. (1994). *The Balance of Nature?* Chicago, IL: University of Chicago Press.

PIPER, J.K. (1995). Composition of prairie plant communities on productive versus unproductive sites in wet and dry years. *Can. J. Bot.* **73**, 1635-1644.

PIPER, J.K. (1999). Natural systems agriculture. In: *Biodiversity in Agroecosystems*. (COLLINS, W.W. & QUALSET, C.O., eds.), pp. 167-196. Boca Raton, Florida: CRC Press.

PIPER, J.K. & PIMM, S.L. (2001, in press). The creation of diverse prairie-like communities. *Restor. Ecol.*

RASTETTER, E.B., GOUGH, L., HARTLEY, A.E., HERBERT, D.A., NADELHOFFER, K.J. & WILLIAMS, M. (1998). A revised assessment of species redundancy and ecosystem reliability. *Conserv. Biol.* **13**, 440-443.

ROBINSON, J.V. & DICKERSON, J.E. JR. (1987). Does invasion sequence affect community structure? *Ecology* **68**, 587-595.

SAMUELS, C.L. & DRAKE, J.A. (1997). Divergent perspectives on community convergence. *Trends Ecol. Evol.* **12**, 427-432.

SAMUELS, C.L. & KOT, M. (2001, in review). A new approach for modeling succession. *J. Theor. Biol.*

STEPHENS, G.R. & WAGGONER, P.E. (1970). The forests anticipated from 40 years of natural transitions in mixed hardwoods. *Conn. Agric. Exp. Sta. Bull.* **707**, 1-58.

STEPHENS, G.R. & WAGGONER, P.E. (1980). A half century of natural transitions in mixed hardwood forest. *Conn. Agric. Exp. Sta. Bull.* **783**, 1-43.

TILMAN, D. (1996). Biodiversity: population versus ecosystem stability. *Ecology* **77**, 350-363

TILMAN, D., LEHMAN, C.L. & BRISTOW, C.E. (1998). Diversity-stability relationships: statistical inevitability or ecological consequence? *Am. Nat.* **151**, 277-282.

TILMAN, D. & DOWNING, J.A. (1994). Biodiversity and stability in grasslands. *Nature* **367**, 363-365.

USHER, M.B. (1981). Modelling ecological succession, with particular reference to Markovian models. *Vegetatio* **46**, 11-18.

USHER, M.B. (1992). Statistical models of succession. In: *Plant Succession: Theory and Prediction.* (GLENN-LEWIN, D.C., PEET, R.K. & VEBLEN, T.T., eds.), pp. 215-248. London: Chapman and Hall.

WAGGONER, P.E. & STEPHENS, G.R. (1970). Transition probabilities for a forest. *Nature* **225**, 1160-1161.

WALKER, B.H. (1992). Biodiversity and ecological redundancy. *Conserv. Biol.* **6**, 18-23.

WILBUR, H.M. & ALFORD, R.A. (1985). Priority effects in experimental pond communities: responses of *Hyla* to *Bufo* and *Rana*. *Ecology* **66**, 1106-1114.

YACHI, S. & LOREAU, M. (1999). Biodiversity and ecosystem productivity in a fluctuating environment: the insurance hypothesis. *Proc. Natl. Acad. Sci. USA* **96**, 1463-1468.

APPENDIX

APPENDIX

JAVA SOURCE CODE.

This appendix contains source code for the Java program used to iterate and find asymptotic distributions for Markov set-chains. The program comprised three Java classes, each given separately. The first contains matrix operations used in the main program. The second is the main program that performs the Hi-Lo method. The third contains all input/output related methods for reporting results. All were written and run using Microsoft Visual J++, version 6.0.

Matrix operations

```
/*********************************************************************
This is a class of the linear algebra operations used by Fuzzy, the
main program that iterates Markov set-chains. It also includes methods
to read and display matrices.
*********************************************************************/
// copy a vector
// copy a matrix
// sum components of a vector
// row sum a matrix
// take inner product of two vectors
// multiply vector*matrix
// multiply matrix*vector
```

```
// multiply matrix*matrix
// read a matrix of doubles from a text file
// read a vector of doubles from a text file
//*************************************************************************

import java.io.*;
public class MatrixMath {

//*************************************************************************

// copy a vector
  public static double[] matrixCopy(double todo[]){
     int m = todo.length-1;
     double done[] = new double[m+1];
     for(int i =1;i<=m;i++){
        done[i] = todo[i];}
     return done;}

//*************************************************************************

// copy a matrix
  public static double[][] matrixCopy(double[][] todo){
     int m = todo.length-1;
     int n = todo.length-1;
     double done[][] = new double[m+1][n+1];
     for (int i = 1; i<=m; i++){
        for (int j=1; j<=n;j++){
```

```
            done[i][j] = todo[i][j];}}
        return done;}

//*********************************************************************

// sum rows of a mxn matrix into a vector of length m
    public static double[] matrixRowSum(double  matrix[][]){
       int m = matrix.length-1;
       int n = matrix[0].length-1;
       double done[] = new double[m+1];
       for (int i =1;i<=m;i++){
          double sum = 0;
          for (int j=1;j<=n;j++){
             sum = sum + matrix[i][j];}
          done[i] = sum; }
       return done;}

//*********************************************************************

// sum a vector row
    public static double vectorSum(double vector[]){
       int m = vector.length-1;
       double sum = 0;
       for (int i =1;i<=m;i++){
          sum = sum + vector[i];}
       return sum;}
```

```
//*********************************************************************

// multiply vector*vector
  public static double matrixMultiply (double[] vector1, double[]
    vector2){
    int vector_length = vector1.length-1;
    int m = vector2.length-1;
    double temp;double done = 0;
    if (vector_length == m){
         temp = 0;
         for (int i = 1; i<=m;i++){
           temp = temp + vector1[i]*vector2[i];
         done = temp;} }
    else System.out.print("Indices don't match!!!");
    return done;}

//*********************************************************************

// multiply vector*matrix
  public static double[] matrixMultiply ( double[] vector, double[][]
    matrix){
    int vector_length = vector.length-1;
    int m = matrix.length-1;
    int n = matrix[0].length-1;
    double done[] = new double[n+1];
    double temp;
    if (vector_length == m){
```

```
      for (int j = 1;j<=n;j++){
        temp = 0;
        for (int i = 1; i<=m;i++){
          temp = temp + vector[i]*matrix[i][j];}
        done[j] = temp;}}
    else System.out.print("Indices don't match!!!");
    return done;}

//*********************************************************************

// multiply matrix*vector
  public static double[] matrixMultiply (double[][] matrix, double[]
    vector){
    int vector_length = vector.length-1;
    int m = matrix.length-1;
    int n = matrix[0].length-1;
    double done[] = new double[m+1];
    double temp;
    if (vector_length == n){
      for (int i = 1;i<=m;i++){
        temp = 0;
        for (int j = 1; j<=n;j++){
          temp = temp + matrix[i][j]*vector[j];}
        done[i] = temp;}}
    else System.out.print("Indices don't match!!!");
    return done;}
```

```
//*********************************************************************

// multiply matrix*matrix
  public static double[][] matrixMultiply (double[][] matrix1,
    double[][] matrix2){
    int m1 = matrix1.length-1;
    int n1 = matrix1[0].length-1;
    int m2 = matrix2.length-1;
    int n2 = matrix2[0].length-1;
    double done[][] = new double[m1+1][n2+1];
    double temp;
    if (n1 == m2){
      for (int i = 1;i<=m1;i++){
          for (int j = 1; j<=n2;j++){
            temp = 0;
            for(int k = 1;k<=n1;k++){
              temp = temp + matrix1[i][k]*matrix2[k][j];}
            done[i][j] = temp;}}}
    else System.out.print("Indices don't match!!!");
    return done;}

//*********************************************************************
```

```
// matrix input routine for doubles
  public static double[][] getDoubleMatrix (String fileName){
  double mtx[][];
  int rows; int columns;{
    try{
      FileReader fr = new FileReader(fileName);
      BufferedReader br = new BufferedReader(fr);
      StreamTokenizer st = new StreamTokenizer(br);
      st.whitespaceChars(',',',');
      st.nextToken(); // advance to the next token
      rows = (int)st.nval; // grab the next token
      st.nextToken();
      columns = (int)st.nval;
      mtx = new double[rows+1][columns+1];
      for (int i=1;i<=rows;i++){
        for (int j=1;j<=columns;j++){
          st.nextToken();
          mtx[i][j] = st.nval;}}
      fr.close();}
    catch(Exception e) {
      System.out.println("exception:  "+ e);
      mtx = new double[1][1];}
  }return mtx;}

//*********************************************************************
```

```
// matrix input routine for double vectors
   public static double[]getDoubleVector (String fileName){
   double mtx[];
   int columns;{
      try{
         FileReader fr = new FileReader(fileName);
         BufferedReader br = new BufferedReader(fr);
         StreamTokenizer st = new StreamTokenizer(br);
         st.whitespaceChars(',',',',');
         st.nextToken(); // advance to the next token
         columns = (int)st.nval; // grab the next token
         mtx = new double[columns+1];
            for (int j=1;j<=columns;j++){
               st.nextToken();
               mtx[j] = st.nval;}
         fr.close();
      }
      catch(Exception e) {
         System.out.println("exception:  "+ e);
         mtx = new double[1];}}
return mtx;}}
```

Hi-Lo method for iterating Markov set-chains

```
/**********************************************************************
This is a class that iterates Markov set-chains.  It takes an initial
high and low (P and Q) matrix, checks to see whether they are tight,
if necessary tightens them, and then calculates the column-tight-
component bounds at each step until the system converges to the
specified number of decimal places.
**********************************************************************/

import java.io.*;
public class fuzzy extends MatrixMath{

//**********************************************************************

double P[][], Q[][], L[][], H[][], sl[], sh[], L2[][], H2[][],
fixed1[][], fixed2[][], safeP[][], safeQ[][], l[], h[], p[], q[],
lcalc[], hcalc[], lo[], hi[], s[], t[];
int n, m, rankl[], rankh[];

public static void main(String args[]){
//get files and create the object to work on:
   fuzzy f = new fuzzy();
   f.P = MatrixMath.getDoubleMatrix("Input Files/T2P.txt");
   f.Q = MatrixMath.getDoubleMatrix("Input Files/T2Q.txt");
   f.s = MatrixMath.getDoubleVector("Input Files/T2s.txt");
   f.t = MatrixMath.getDoubleVector("Input Files/T2t.txt");
```

```
f.n = f.P.length-1; // determine the size of matrix P
f.m = f.Q.length-1; // determine the size of matrix Q
f.safeP = MatrixMath.matrixCopy(f.P); //save copies of P and Q.
f.safeQ = MatrixMath.matrixCopy(f.Q);

//**********************************************************************

//Tighten and record tightening info:
  //open and start tightness output file
  fuzzy.startFile("Output Files/tightout.txt");
  f.P = MatrixMath.matrixCopy(fuzzy.intTighten (f.safeP, f.safeQ,
    true, 4)); //give tightened P
  f.Q = MatrixMath.matrixCopy(fuzzy.intTighten(f.safeQ, f.safeP,
    false, 4 )); //give tightened Q
  FuzzOut.MatrixPrt(f.safeP, "Initial P", "Output Files/tightout.txt",
    1);
  FuzzOut.MatrixPrt(f.safeQ, "Initial Q", "Output Files/tightout.txt",
    1);
  FuzzOut.MatrixPrt(f.P,"Tightened P","Output Files/tightout.txt",1);
  FuzzOut.MatrixPrt(f.Q,"Tightened Q","Output Files/tightout.txt",1);
  f.L = MatrixMath.matrixCopy(f.P);   //set L and H to the initial
    tightened P and Q.
  f.H = MatrixMath.matrixCopy(f.Q);
```

```
//Iterate and check for convergence:
   double check = 1; //start the convergence test
   int time = 1; //start the iterations clock.
   int printat = 1;
   FuzzOut.Lout(f.L, f.H, "itermat.txt", time); //output starting
     matrices to a file.
   while (check >= .0000001){ //set convergence level.

//initialize variables for rankings
        f.p=new double[f.n+1];
        f.q=new double[f.n+1];
        f.lcalc=new double[f.n+1];
        f.hcalc=new double[f.n+1];
        f.L2=new double[f.n+1][f.n+1];
        f.H2=new double[f.n+1][f.n+1];
        f.lo=new double[f.n+1];
        f.hi=new double[f.n+1];

//get and display rankings
     for (int j=1; j<=f.n;j++){
        f.l=new double[f.n+1];
        f.h=new double[f.n+1];
        for (int i = 1;i<=f.n;i++){ f.l[i] = f.L[i][j];}
        for (int i = 1;i<=f.n;i++){ f.h[i] = f.H[i][j];}
        f.rankl = fuzzy.rankColumn(f.l);
        f.rankh = fuzzy.rankColumn(f.h);
```

```
//find the lo and hi vector for each element of jth row
      for (int i = 1;i<=f.n;i++){
        for (int r=1;r<=f.n;r++){ f.p[r] = f.P[i][r];}
        for (int r=1;r<=f.n;r++){ f.q[r] = f.Q[i][r];}
        for (int r=1;r<=f.n;r++){ f.lcalc[r] = f.L[r][j];}
        for (int r=1;r<=f.n;r++){ f.hcalc[r] = f.H[r][j];}
        f.lo = fuzzy.loVector(f.p,f.q,f.rankl);
        f.L2[i][j] = MatrixMath.matrixMultiply(f.lo,f.lcalc);

        //calculate element i,j of new matrix.
        f.L2[i][j] = numRound(f.L2[i][j],4);
        for (int r=1;r<=f.n;r++){ f.p[r] = f.P[i][r];}//reset p and q
        for (int r=1;r<=f.n;r++){ f.q[r] = f.Q[i][r];}

        f.hi = fuzzy.hiVector(f.p,f.q,f.rankh);
        f.H2[i][j] = MatrixMath.matrixMultiply(f.hi,f.hcalc);

        //calculate element i,j of new matrix.
        f.H2[i][j] = numRound(f.H2[i][j],4);}}
  time++;
  check=0;
  for (int i=1;i<=f.n;i++){
    for (int j=1;j<=f.n;j++){
      check=check+Math.abs(f.L2[i][j]-f.L[i][j]);}  }
  if (printat>=1)
  {FuzzOut.Lout(f.L2, f.H2,"itermat.txt", time);
```

```
   printat=0;

//deal with initial conditions to predict distribution at each time
   double s1lo[]=new double [f.n+1];
   double s1hi[]=new double [f.n+1];

//start with all in state 1, or remove comment marks from below and
   put in desired values.
      f.s[1]=0.522;
      f.t[1]=0.692;
   //f.s[2]=0.024;
   //f.s[3]=0.023;
   //f.s[4]=0.538;
   //f.s[5]=0.092;
   //for (int i=2;i<=f.n;i++){
      // f.s[i]=0.308;
      // f.t[i]=0.478;
   //}
   //f.t[1]=0.103;
   //f.t[2]=0.035;
   //f.t[3]=0.162;
   //f.t[4]=0.74;
   //f.t[5]=0.297;
   */
//}
```

```
  for (int j=1; j<=f.n;j++){

//get and display rankings
        f.sl = new double[f.n+1];
        f.sh = new double[f.n+1];
        for (int i = 1;i<=f.n;i++){ f.sl[i] = f.L[i][j];}
        for (int i = 1;i<=f.n;i++){ f.sh[i] = f.H[i][j];}
        f.rankl = fuzzy.rankColumn(f.sl);
        f.rankh = fuzzy.rankColumn(f.sh);

//find the lo and hi vector for jth row
        //for (int i = 1;i<=f.n;i++){
          for (int r=1;r<=f.n;r++){ f.p[r] = f.s[r];}
          for (int r=1;r<=f.n;r++){ f.q[r] = f.t[r];}
          for (int r=1;r<=f.n;r++){ f.lcalc[r] = f.L[r][j];}
          for (int r=1;r<=f.n;r++){ f.hcalc[r] = f.H[r][j];}
          f.lo = fuzzy.loVector(f.p,f.q,f.rankl);
          s1lo[j] = MatrixMath.matrixMultiply(f.lo,f.lcalc);

//calculate element i,j of new matrix.
          s1lo[j] = numRound(s1lo[j],4);
          for (int r=1;r<=f.n;r++){ f.p[r] = f.s[r];}

//reset p and q.
          for (int r=1;r<=f.n;r++){ f.q[r] = f.t[r];}
          f.hi = fuzzy.hiVector(f.p,f.q,f.rankh);
```

```
            s1hi[j] = MatrixMath.matrixMultiply(f.hi,f.hcalc);

//calculate element i,j of new matrix.
            s1hi[j] = numRound(s1hi[j],4);
            //}
    }
    //s1lo=f.matrixMultiply(f.s,f.L2);
    //s1hi=f.matrixMultiply(f.s,f.H2);

    FuzzOut.FuzzStatesOut(s1lo,s1hi,"states.txt",time);
    System.out.println ("Step " + (time-1) + "\tConvergence check =
       "+check);}
    printat++;
    f.L = MatrixMath.matrixCopy(f.L2);
    f.H = MatrixMath.matrixCopy(f.H2);}
    System.out.println("System converged at time = " + (time-1));}
```

```
//*********************************************************************
//This section contains methods used by Fuzzy.
//*********************************************************************

//intTighten checks tightness
//numRound rounds numbers to a specified number of decimal places
//rankColumn returns rankings for the elements of a vector
//loVector finds the lo vector
//hiVector finds the hi vector
//startFile creates a new output file with specified filename

//*******************************************************

//Method that checks to see whether the interval is tight for a row.
  public static double [][] intTighten(double tofix1[][],double
    tofix2[][],boolean totight, int decimals){
    int n = tofix1.length-1;
    //make copies of the matrices to be tightened
    double fixed1 [][]= new double [n+1][n+1];
    double fixed2 [][]= new double [n+1][n+1];
    double rowsums2[] = MatrixMath.matrixRowSum(tofix2);
    if (totight){ // true means P is being tightened.
      for (int i=1; i<=n;i++){
        for (int j=1; j<=n; j++){
          if (rowsums2[i]- tofix2[i][j] + tofix1[i][j]<1){
            double num=numRound(rowsums2[i]-tofix2[i][j], decimals);
            fixed1[i][j] = 1-num;
```

```
            FuzzOut.TightOut(fixed1[i][j],i,j);}
            else {fixed1[i][j]=tofix1[i][j];}
            fixed1[i][j] = numRound(fixed1[i][j],decimals);}}
      return fixed1;}
      else { // Q is being tightened.
        for (int i=1; i<=n;i++){
          for (int j=1; j<=n; j++){
            if (rowsums2[i]- tofix2[i][j] + tofix1[i][j] > 1){
              double num=numRound(rowsums2[i]-tofix1[i][j],decimals);
              fixed2[i][j] = 1-num;
            FuzzOut.TightOut(fixed2[i][j],i,j);}
            else {fixed2[i][j]=tofix1[i][j];}
            fixed2[i][j] = numRound(fixed2[i][j],decimals);}}
      return fixed2;}  }

//*********************************************************************

//method to round a number to the desired number of decimal places
   public static double numRound (double num, int decs){
      int places=1;
      for(int i=1; i<=decs+1; i++){
        places = places*10;}
      num = Math.round(num*places);
      num = num/places;
      return num; }

//*********************************************************************
```

```
//Method that returns rankings for element of a vector.
  public static int[] rankColumn(double col[]){
    int n = col.length-1;
    int rank[] = new int[n+1];
    for(int i=1;i<=n;i++){
        double temp = 27;
        for(int k = 1;k<=n;k++){
          if(col[k]<=temp){
            rank[i] = k;
            temp = col[k];   }}
        col[rank[i]] = 2727;}
    return rank;   }

//*********************************************************************

//Method that returns the lo vector.
  public static double[] loVector(double pvec[], double qvec[], int
    rank[]){
    int t = pvec.length-1;
    double lo[] = qvec;
    //replace high-ranking elements until sum is <1.
    while (vectorSum(lo) > 1){
      lo[rank[t]] = pvec[rank[t]];
      t--;}
```

```
    //if the p vector doesn't sum to 1, adjust last changed element.
    if (vectorSum(lo)<1) {
      //In case the q values didn't sum to more than one anyway.
      if (t == pvec.length-1){t=pvec.length-2;}
      lo[rank[t+1]] = 1- (vectorSum(lo)- lo[rank[t+1]]);}
    return lo;}

//*********************************************************************

//Method that returns the hi vector.
  public static double[] hiVector(double pvec[], double qvec[], int
    rank[]){
    int s = qvec.length-1;
    double hi[] = pvec;
    //replace high-ranking elements until sum is <1.
    while (vectorSum(hi) < 1){
      hi[rank[s]] = qvec[rank[s]];
      s--;}
    //if p vector doesn't sum to 1, adjust the last changed element.
    if (vectorSum(hi)>1) {
      //In case the q values didn't sum to more than one anyway.
      if (s == pvec.length-1){s=pvec.length-2;}
      hi[rank[s+1]] = 1- (vectorSum(hi)- hi[rank[s+1]]);}
    return hi;}

//*********************************************************
```

```
//Method that starts a new output file with the given filename
  public static void startFile (String fname){
  FileWriter fw; // starts the file output object for process data
    BufferedWriter out;
    try  {
      fw = new FileWriter("Output Files/tightout.txt");
      out = new BufferedWriter(fw);
      out.write("Element: \tRow: \tTightened to: \n");
      out.close();}
     catch (Exception e){
      System.err.println ("Error writing to tightness data file");}}}
```

Input and output methods

```
//*********************************************************************
This class contains methods for output of the 'fuzzy' class for
iterating Markov set-chains.
**********************************************************************/

//MatrixPrt writes matrix to a specified text file.
//Lout writes L and H matrices for each time step to a file.
//FuzzStatesOut writes distribution of individuals among states at
   each time to a text file
//TightOut writes tightness and tightening information to a file.

//*********************************************************************

import java.io.*;
public class FuzzOut extends fuzzy{

//*********************************************************************

// a method that prints a given matrix a file to a given file name
  public static void MatrixPrt (double matrix[][], String name, String
     fname, int iter){
  FileWriter fw; // declare a file output object
  BufferedWriter out;
    int n = matrix.length-1;
```

```
try{
   if (iter<=0){
      fw = new FileWriter(fname);}
   else {
      fw = new FileWriter(fname, true);}
   out = new BufferedWriter(fw);
   out.write("\nMatrix: " +name+ "\n");
    int i = 1;
    while (i<=n){
       int j = 1;
       while (j<=n){
         out.write("\t" + matrix[i][j]);
         j++;}
      out.write ("\n");
      i++;
      }    out.close();}
 catch (Exception e){
     System.err.println ("Error writing to tightness data file");}}

//*****************************************************************
```

```
// a method that writes L and H matrices for each time step to a file
  public static void Lout(double Lnext[][], double Hnext[][], String
    fname, int iter){
  int n = Lnext.length-1;
  FileWriter fw; // declare a file output object
  BufferedWriter out;
  try{
    if (iter==1){
      fw = new FileWriter(fname);} // Make new file
    else {
    fw = new FileWriter(fname, true);} // Append to existing file
    out = new BufferedWriter(fw);
    //out.write ("\n\tLow Matrix:\t\t\t\t\t\t\tHigh Matrix,
      \t\t\t\t\t\t\tDistribution:\n");
    out.write(""+iter);
     int i = 1;
     while (i<=n){
        //out.write("\t");
        int j = 1;
        while (j<=n){
          out.write("\t"+ Lnext[i][j]);
          j++;}
        out.write("\t");
        j = 1;
        while (j<=n){
          out.write("\t"+ Hnext[i][j]);
          j++;}
```

```
        out.write ("\n");
        i++;}
      /* out.write("\nHigh Matrix, Time = " +iter+ ":\n");
       i = 1;
       while (i<=n){
          int j = 1;
          while (j<=n){
            out.write("\t" + Hnext[i][j]);
            j++;}
          out.write ("\n");
          i++;}
    */
       out.close();}
    catch (Exception e){
      System.err.println  ("Error  writing  iterated  matrices  to  a
        file");}}

//*************************************************************************
```

```
// a method that writes L and H matrices for each time step to a file
public static void FuzzStatesOut(double LoStates[],double
HiStates[],String fname,int time){
  int n = LoStates.length-1;
  FileWriter fw; // declare a file output object
  BufferedWriter out;
  try {
    if (time==2){
      fw = new FileWriter(fname); // Make new file
        out = new BufferedWriter(fw);
    else {
      fw = new FileWriter(fname, true); // Append to existing file
      out = new BufferedWriter(fw);}
    out.write("" + (time-1));
    int j = 1;
    while (j<=n){
      out.write("\t"+ LoStates[j]);
      j++;}
       out.write("\t");
     int k=1;
     while (k<=n){
      out.write("\t"+ HiStates[k]);
      k++;}
     out.write("\n");
     out.close();}
  catch (Exception e){
      System.err.println ("Error writing states to a file");}}
```

```
//*********************************************************************

// a method that writes the data on what was tightened to a file
called "tightout.txt"
  public static void TightOut(double tightened, int row, int col){
  FileWriter fw; // declare a file output object
  BufferedWriter out;
    try{
     fw = new FileWriter("Output Files/tightout.txt", true);
     out = new BufferedWriter(fw);
       out.write(col + "\t" + row + "\t" + tightened + "\n");
     out.close();}
  catch (Exception e){
       System.err.println ("Error writing to tightness data file");}}
```

VITA

Corey L. Samuels was born in Duluth, Minnesota on June 11, 1969. As a young woman she lived with her family in Berlin, Germany, where she graduated from JFKS German-American Community School in 1986. Her undergraduate studies strung across nearly as many different continents as academic majors, taking her from Germany to a year of work-study on a kibbutz in Israel, and then back to the United States where she studied Chemical Biology at Stevens Institute of Technology in Hoboken, New Jersey. She finally landed at the University of Arizona, where she completed a Bachelor of Arts degree in Political Science in 1991. In 1992, she began an internship in sustainable agriculture at The Land Institute in Salina, Kansas. The ten-month internship flowed into a staff position, and fostered a growing fascination with modeling and visualizing complexity. Pursuing this interest, in 1994 Corey relocated to Knoxville, Tennessee to begin doctoral studies in Ecology at the University of Tennessee. In the summer of 2000, she changed venue to Seattle, Washington for her final months of dissertation writing. Corey now resides in Seattle, and the only thing she knows with certainty about her future is that it will be an adventure.